Classification in multi- observational setting using latent Gaussian processes

Sudhanshu Nautiyal

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 30.09.2017

Thesis supervisor:

Prof. Samual Kaski

Thesis advisor:

M.Sc. Sami Remes

Aalto University
School of Science

AALTO UNIVERSITY
SCHOOL OF SCIENCE

ABSTRACT OF THE
MASTER'S THESIS

Author: Sudhanshu Nautiyal

Title: Classification in multi- observational setting using latent Gaussian processes

Date: 30.09.2017 Language: English Number of pages: 6+46

Computer Science and Information Science

Professorship: Machine Learning and Data Mining

Supervisor: Prof. Samual Kaski

Advisor: M.Sc. Sami Remes

Widespread interest in the usage of data collection devices all around the world has resulted in an increasingly large number of sequential multivariate datasets. Be it IoT applications, wearable sensors, medical records or fMRI records number of datasets with series of multiple observations per sample is growing. Most of these datasets typically constitute observations of a fairly complex process and contain thousands of data points. High dimensionality of these datasets combined with their susceptibility to missing data and multi-observational setting can make implementing traditional data analysis techniques for these datasets challenging.
Impressed with their ability to propagate prior information about latent processes and learn the components nonparametrically, we explore Bayesian latent variable models and propose a multi-observational sparse Gaussian process based classifier that can efficiently classify observations by learning separate latent space representation for each observation. As a precursor to the development of our proposed model we derived a scalable variational approximation for the semiparametric latent factor model and further extended it to accommodate multi-observational datasets.
Finally, we perform several experiments and demonstrations with artificial datasets on the proposed model to ensure that model is not overly sensitive to the variability of parameters and can achieve classification performance at-par with other popular classification methods.

Keywords: For keywords choose concepts that are central to your thesis

Acknowledgements

This thesis was done at the department of computer science and information science in Aalto university school of science.

I would first like to thank Prof. Samual Kaski for his able guidance, encouragement and opportunity to work in the inspiring environment of probabilistic machine learning research group. I am also deeply grateful to my instructor Msc. Sami Remes for his suggestions and helpful discussions. I have learned a lot with our regular discussions and Sami's insights into GP models have been extremely beneficial with regards to this work.

I am also thankful to Liisa Puurunen and Jussi Kulla for their help and constant motivation over the course of this work.

Last but not the least, I wish to thank my parents for their love and unwavering support in the entirety of my life, I shall remain indebted to them forever.

Helsinki, 1.10.2017

Sudhanshu Nautiyal

Contents

Symbols and abbreviations

Symbols

$\mathcal{N}$	Normal distribution
$\mathcal{TN}$	Truncated Normal distribution
$\mathcal{G}$	Gamma distribution
$\mathbb{R}$	Set of all real numbers
I	Identity matrix

Operators

$A \otimes B$	Kronecker product between A and B
$\frac{\mathrm{d}}{\mathrm{d}t}$	derivative with respect to variable t
$\sum_i^N$	sum over index i till N
$\prod_i^N$	product over index i till N
$p(Y\|X,Z)$	probability distribution of Y given X and Z
$\mathbb{E}[u]$	expectation of u
$\delta(e)$	Kronecker delta: 1 if e is true otherwise 0

Abbreviations

GP	Gaussian Process
MCMC	Markov Chain manto Carlo
VB	Variational Bayes
KL	Kullback-Libler
RHS	Right Hand Side
ELBO	Evidence Lower Bound
FA	Factor Analysis
PCA	Principal Component Analysis
ICA	Independent Component Analysis
LVM	Latent Vector Model
IVM	Informative Vector Machine
LCPC	Latent Gaussian Process Classifier
LDA	Linear Discriminat Analysis
SVM	Support Vector Machine
GPLVM	GP Latent Vector Model

1 Introduction

1.1 Research Context

In multi observational datasets each instance consists of multiple sequences of observations. fMRI readings, medical history of personals, sensor outputs from wearable devices etc are some examples where each instance would have multiple data sequences attached with it. Most of these datasets typically are observations of a fairly complex process and contain thousands of data points. Data analysis in these multi observational settings has some peculiar challenges. For example, in IoT applications there can be hundreds of sensors that generate data of extremely high dimensionality. Moreover, there can be missing parts at separate locations in the sensor output of each instance making traditional data analysis techniques harder to impalement.

Due to their many desirable qualities like ability to provide uncertainty around the estimations, seamless incorporation of new information to the model, handling missing data etc. Bayesian methods provide a principled path towards performing statistical analysis when dealing with complex phenomenon. Systematic approach of moving from assumptions to inference has made Bayesian methods popular among statistician and modelers. Moreover, in this quest of making sense from a complex and overly high dimensional datasets, non parametric models are of particular interest due to their propensity to make least amount of assumptions about the underlying structure in data. Gaussian Processes are one of the extremely well studied Bayesian non parametric models with a century old history in statistical inference. They not only allow us to put informative priors over the data generating functions through kernels but also make it easier to handle missing data by mapping observations to continuous domain. However, inference process in GP constitute large matrix inversions and thus can make GP models prone to scalability issues, specially with large datasets. Additionally, complex GP models might yield intractable posteriors and require suitable approximation schemes to produce meaningful solutions.

Latent Variable models assume that the observations are generated through interactions between some hidden variables which if found can explain much of the variability in the observations.These latent variables can represent hidden commonalities across observations and thus are usually lesser in the number than actual data dimensions. Their ability to explain away the data variability with fewer components and capability to give an insight into data generation process makes latent variable models a practical choice for the analysis of high dimensional datasets. Moreover, projection of original observations into a compact latent space before prediction can also increase the efficiency of a prediction model. However, it can be challenging to incorporate the prior knowledge about latent functions in standard latent variable models.

In recent years Gaussian process based latent variable models have gained widespread interest in machine learning community due to their ability to propagate prior information about latent process and learn the components nonparametrically. However, even in these models it can be awkward to do so in an observation specific manner and learn separate latent space for each observation.

Our objective in this thesis thus, is to develop a scalable Gaussian process based latent variable model that would not only be able to handle multi observational setting but also use the the external knowledge in terms of classification labels to learn better latent space projection techniques.

1.2 Thesis Contributions

This thesis presents a Gaussian process based classification model in a multiple observational setting where each observation consists of multivariate data sequences and corresponding category label. The model projects each high dimensional observation into its own latent space using Gaussian processes in order to extract richer representations that might help in achieving effective classification results. In order to maintain the scalability of the process, a sparse approximation to the posterior of proposed model is also presented.

As a precursor to the proposed model, we also derive a sparse variational approximation to GP based factor analysis model closely related to the semi parametric latent factor model proposed by [Seeger, Teh, et al. 2004]. We further extend latent factor model to handle mult-observational data.

Experiments and demonstrations with artificial datasets are performed to ensure that derived model is not overly sensitive to the variability of parameters and achieves classification performance at par with other popular classification methods.

1.3 Structure of the thesis

This thesis is organised as follows: Chapter 2 provides necessary background information by first discussing Bayesian Inference as well as parametric and non parametric models. After a brief look at the features of non parametric models, Gaussian processes are introduced as one of the prime examples of Bayesian nonparametrics. Since more often than not GP posteriors can become intractable we conclude this chapter with discussions on approximate inference techniques and their application on Gaussian processes.

Chapter 3 explores latent variable models and reviews probabilistic factor analysis as the precursor to more involved Gaussian process based factor analysis model. Model's variational approximation and demonstration on an artificial dataset are provided next. We further extend the model to include multiple observational setting in section 3.3 and finally end the chapter with a discussion on our extension and corresponding demonstrations.

Chapter 4 introduces latent Gaussian process based classifier by combining the extended Gaussian process based factor analysis model with a classifier. Full variational well as sparse approximation of the model is presented along with a demonstration on artificial dataset.

Chapter 5 presents experimental results on artificial as well as a benchmark datasets and finally, Chapter 6 concludes the thesis by first contemplating some related works and then discussing on final remarks on the model, underlining few pointers to further research.

Some of the technical derivation related to the presented models are provided in Appendices.

2 Background

This chapter contains a review of various fundamental concepts that will be used later in the thesis as basic building blocs for the more complex models. We briefly revise the bayesian inference and then explore inference problem from a non parametric perspective in section 2.1. Later, section 2.2 provides a small introduction to Gaussian Processes as an important bayesian nonparametric method and finally the chapter concludes with a short primer on sparse approximation techniques for Gaussian processes in section 2.3.

2.1 Bayesian Inference

A typical statistical inference problem usually deals with estimation of unknown quantities based on some observed information. Due to their many desirable qualities like ability to provide uncertainty around the estimations, seamless incorporation of new information to the model and elimination, Bayesian methodology has considerable advantage over other approaches to data analysis and statistical inference.

Bayesian methods approach the solution of inference problem by setting up a model that constitutes our assumptions and knowledge about all relevant observed as well as unobserved quantities as probabilistic statements. The inference process in Bayesian models consists of three steps: first formulation of initial knowledge about the model and its constituents in terms of prior and joint distribution, application of Bayes' theorem to figure out the posterior - conditional probability distribution of unobserved quantities given the observed ones and finally a validation step is performed to determine model's correctness. The posterior distribution derived during inference process is considered to be the underlying distribution from which all unobserved quantities were generated in the past and will continue to do so in future while the validation step formally determines model's ability to explain the observations [Gelman, J. B. Carlin, et al. 2014].

In a generic Bayesian analysis process, the particular data generating model we believe the observations arose from are described through model-parameters and thus, the objective of inference process revolves around finding the possible values of parameters that could generate the given observations. If M denotes our model and θ its parameters,

$$p(\theta \mid observations, M) \propto p(observations \mid \theta, M)p(\theta, M) \tag{1}$$

Here the first term at RHS tells us the probability of seeing those observations given a particular set of parameters i.e. likelihood and the second term incorporates our beliefs about those parameter values before seeing the observations, i.e. the prior. One the posterior is identified, correct data generating model along with its parameters is considered to be recognized and one could use them to either generate or predict new data:

$$p(x \mid \theta, observations, M) = p(x \mid \theta, observations, M)p(\theta \mid observations, M) \tag{2}$$

2.1.1 Models

As alluded in previous section, we usually make certain assumptions about the model based on the specific problem at hand. These salient properties of model are incorporated in the inference process as the model parameters. For example, it is common to assume sampling distribution of a continuous quantity to be a normal distribution. In that case mean and variance would be termed as model parameters. Similarly, number of clusters would be a parameter in case of a mixture model. Values if these specific parameters then can be used describe that particular model and hence once the value of these parameters are known, model is said to be identified, i.e. complete knowledge about observed data and its generating process is transferred in the parameters from observations, or in other words, parameters now contain 'everything there was to know about data relevant to future predictions' [Ghahramani 2013]. From the perspective of equation 2,

$$p(x \mid \theta, observations, M) = p(x \mid \theta, M)$$

However, this requirement of making initial assumption about parameters implies that the underlying distribution for these models is restricted to be in the same category as the other models with similar parameterization [Kendall et al. 1994]. This restriction as we will see later might make our models inflexible and of limited capacity.

Conversely, in cases where we want to make minimal or no assumptions about model,a generic model with no fixed number of parameters can be stipulated called a non-parametric model. For example, a non parametric regression model would not consider any specific function form and would just specify a collection of functions one of which might have generated the observations.

Since the model is considered to be the process that generated the observational data during data analysis, another way to look at the distinction between parametric and non parametric models is by considering them to be members of the set of all data generating processes defined in a parameter space where model parameters act as an index such that a vector of model parameters can be used to identify a specific model. A parametric model then will have a finite length parameter vector identifying a subset of data generating process while non parametric model will have infinite number of parameters representing unrestricted amount of models that can be considered.

Apart from possibility to consider a richer set of models by making minimal assumptions, non parametric models also provide a coherent way of model criticism and prior selection. Since prior beliefs in Bayesian methodology must represent the actual state of knowledge without the looking at the observations, traditional model selection approach to alleviate prior uncertainty by selecting the prior with most convincing posterior results (among multiple posteriors derived from different prior contenders) is not only considered incoherent but also unnecessary since appropriately used Bayesian methods should not be overfitting any ways[Rasmussen and Ghahramani 2001, D. J. C. MacKay 1992]. A nonparametric approach however, can go around the the issue of incoherence in models selection as a result of nonparametric

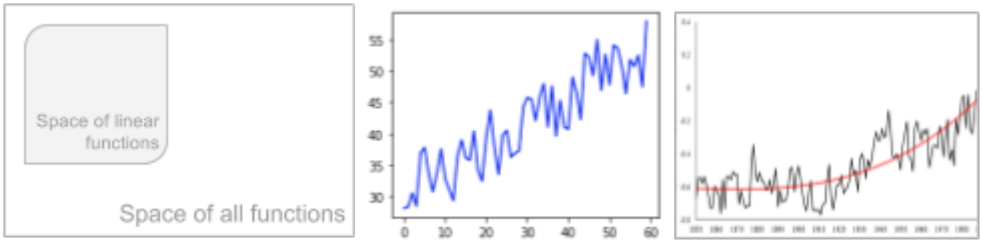

Figure 1: Left: Illustration of function spaces and linear spaces with in it, Middle: A linear trend, Right: Non Linear trend

model's ability to specify unrestricted forms that can accommodate the uncertainty around priors [Hjort 2010].

The problem of choosing a model's parameters beforehand is of prime importance in machine learning arena. Improper model selection can result in an over or under fitted model thereby impacting a model's performance significantly. Moreover, some use cases where identification of underlying structure of data is important (like hidden Markov models, mixture models etc.) might require one to supply the parameters before the inference process has even started. Usually cross validation is used to alleviate these problems [Kohavi 1995]. However, depending upon the situation it might either be inefficient to compare all the possible models or simply incoherent from Bayesian perspective due to re-usage of data multiple times. Bayesian nonparametric methods mitigate this problem by adapting the complexity of model based on available data, [Ferguson 1973, Quintana et al. 2004] i.e allowing for gradually more complex model as the amount of data grows. For example, a traditional mixture model requires number of clusters to be fixed beforehand while a non parametric model would be able to infer even the number of clusters from the data and allows it to increase as it encounters new observations.

2.1.2 Nonparametric models

In most machine learning application our objective is to find a 'pattern' that can be used to explain observed data. A function can be generally used to represent this pattern and thus our objective becomes to search for the function among many possible functions that best explains the data we have observed. Though seemingly infinite, this search space can be significantly reduced by making some common sense assumptions. For example, to find the function that explain clearly linear data in Figure 1(middle) one might want to search within smooth linear function spaces while in case of Figure 1(right) we would like to relax even the linear constraint and search among all possible smooth functions. As we can see the reasoning becomes analogous to putting prior over parameters in Bayesian analysis in the sense that now parameter space represents all possible functions and prior encapsulates our initial guess about the space where our viable solution might exist. (Figure 1). It is clear form above discussion that Bayesian nonparametric model requires us to put probability distribution over these functions. Beginning from Fergusson's work

on Dirichlet's processes [Ferguson 1973] various strategies have been recommended to build these models. [Radford M. Neal 2000] for example, recommends starting from a parametric model and then take infinite limit of it. Since functions are also considered infinite dimensional objects, a lot of work in this field have been done under stochastic process analysis[Orbanz 2009]. Historically Dirichlet's process and Gaussian processes have dominated the nonparametric models literature [Hjort 2010, Rasmussen 1999]. Dirichlet's process is used to put distribution over distributions and hence has been primarily used in relation to density estimation or topic modelling related problem. Gaussian Processes are another famous tool set that have been used extensively to solve Bayesian nonparametric problems where functions are to be estimated, i.e. putting distributions over functions. We will explores Gaussian processes in more detail in next section.

2.2 Gaussian Processes

Due to their seemingly simple nature and interesting analytical properties Gaussian processes have been used extensively in various statistical tasks for decades. For example, [Gelman, J. B. Carlin, et al. 2014] mention them being used for astronomical data modelling in 18th century. Later, significant work on GPs has been done as part of stochastic process literature for example as Wiener processes and later also for time series prediction[Wiener et al. 1964, Papoulis et al. 2002, Kolmogoroff 1941]. In Spatial statistics, following the work of D.G. Krige, gaussian process regression was used for interpolation of values based on space as the input by Matheron in 1973 [Krige 1951, Matheron 1973]. In applied statistics context, Gaussian processes were first used for regression and classification [O'Hagan et al. 1978,C. Williams et al. 1998] and later in a wide variety of applications including density estimation [Leonard 1978, Tokdar n.d.]. Inspired by the link shown between neural networks and Gaussian process by [Kearns et al. 1999], Gaussian processes were quickly extended to machine learning context[C. K. Williams et al. 1996]. With the advent of various scalability techniques [Csató et al. 2002,Seeger, C. K. I. Williams, et al. 2003,E. Snelson et al. 2006] and gains in hardware technology, use of Gaussian process quickly skyrocketed in other areas of machine learning such as, latent vector modelling [Zhao et al. 2009, Neil D Lawrence 2004,Neil D. Lawrence 2005], nonlinear hierarchical learning [Wilson et al. 2012,Damianou et al. 2013] etc.

In the next few sections, we introduce Gaussian process by first providing its definition and reviewing few properties briefly. Later on inference procedure in Gaussian processes is described including examples of regression and classification using GP. Finally, the section concludes with a short note on scalability of gaussian processes.

2.2.1 Introduction

A function is generally thought of as a mapping from an arbitrary set X to another $F(X), X \rightarrow F(X)$. Another interesting way to look at functions can also be of thinking X and F(X) as infinitely long vectors with former's elements acting as an

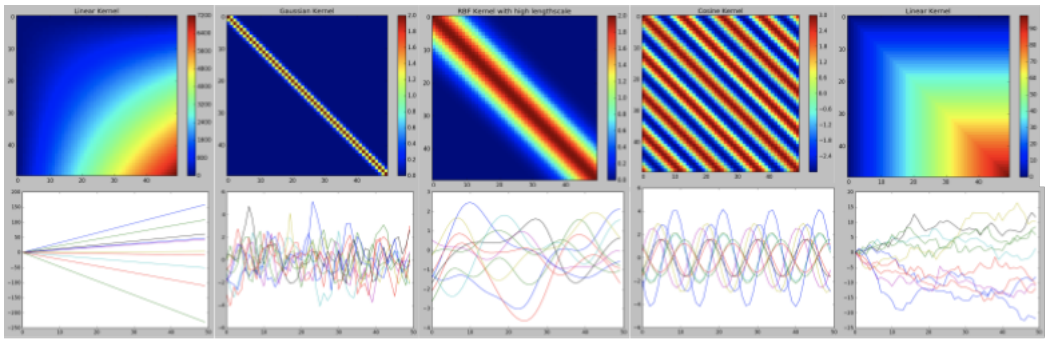

Figure 2: Different Kernels and corresponding sampled GPs

index for the latter, i.e. X as an index set for $F(X)$. A process is then defined as a collection of these random mappings from index set to set of all functions. [Ross 1996].

A Gaussian Process is formally defined as a collection of random variables, any finite set of which is jointly distributed. i.e. for a finite set x contained in X , collection of random variables $F(x)$ will always follow a Gaussian distribution [Rasmussen and Christopher KI Williams. 2006], i.e.

$$p(F|X) \sim \mathcal{N}(m, K) \tag{3}$$

In gaussian process terms f(x) is written as $f(x) \sim GP(m(x), k(x, x))$ where $m(x)$ and $k(x, x)$ are called the mean and covariance function respectively.

From the perspective of functions, one expects that for a given function, data points close/near to each other will have similar values. This notion of nearness can also be seen in general life for example, if one were to measure moisture in environment at a specific time every day for a year we expect days or places near each other to have similar temperatures. Gaussian Process represent this concept of nearness/closeness through a covariance function, i.e. how the function values change with respect to the difference in their index values. A covariance function thus, must take two indices and map them to a real number defining the correlation of underlying function's values at the indices. The matrix K in equation 3 is generated by applying this covariance function $k(x, y) \to \mathbb{R}$ for each pair of elements in X, i.e $K_{ij} = k(x_i, x_j)$.

A covariance function has two important responsibilities. First, when applied pairwise to the elements of set X it must be able to generate a proper covariance matrix (symmetric, positive semi definite etc.) and second, it should also somehow be a measure of nearness between two values, i.e. map its inputs to a real value. Former enables covariance function to significantly determine the properties of to be generated functions like smoothness, length-scale etc. to a large extent while latter means they are same as kernels of SVM literature [Shawe-Taylor et al. 2004]

A Gaussian process is completely determined by its mean and co-variance function and once these two are fixed we can generate sample functions using equation 3 Fig 2 shows various kernels and functions generated by using them as covariance functions for gaussian processes with zero mean.

The mathematical literature on Gaussian process is quite extensive. [Rasmussen and Christopher KI Williams. 2006, Ross 1996] are good references for detailed mathematical properties of GP and of stochastic processes in general. Additionally, [Kendall et al. 1994, Grimmett et al. 2001] are very good references for application of GP in probability theory.

2.2.2 Inference

Under Bayesian nonparametric framework, Gaussian processes are used to express our prior beliefs about underlying function that could describe observed data. By combining these prior assumptions with the observed information we arrive at the posterior representation of the latent function values.

A generic inference process of Gaussian process model with n observations, Y ($Y = [y_1, y_2, \dots y_n]$) at locations X ($X = [x_1, x_2 ... x_n]$) follows a hierarchical structure [Jaakko Riihimäki n.d.]. First, we assume a data generating function $f_i = f(x_i)$ with n latent function values $f_1, f_2 \dots f_n$ such that given these latent values, Y's are independent of locations X.

$$p(Y \mid F) =_{i=1}^{N} p(y_i \mid F_i)$$

The functional space from which these function values can arise is then constrained by putting a suitable GP prior over latent function values.

$$p(F(X) \mid \theta) = GP(m(X), K(X, X' \mid \theta))$$

Finally, we can also put a prior over hyperparameters of covariance function.

$$\theta \sim P(\theta)$$

Now, as explained in the previous section, GP prior corresponds to a multivariate Gaussian distribution and hence prior for F can be written as:

$$p(F \mid X, \theta) = \mathcal{N}(F \mid 0, K_{xx})$$

Where K_{xx} would be the covariance matrix generated through pairwise application of covariance function on observations X. Now following Bayesian inference rules the posterior can be obtained as,

$$p(F \mid X, Y, \theta) = \frac{p(Y \mid F)p(F \mid X, \theta)}{p(Y \mid X, \theta)} \approx \mathbb{N}(F \mid 0, K_{xx}) \prod_{i=1}^{n} p(y_i \mid f_i) \tag{4}$$

Generally, we are also interested in the prediction of new latent values F* (or a new value Y*) on different locations X*. In that case joint distribution of F and F* can be written as,

$$p(F, F^* \mid X, X^*, Y) \sim \mathcal{N}(\begin{bmatrix} F \\ F^* \end{bmatrix} |0, \begin{bmatrix} K_{xx} & K_{x*} \\ K_{*x} & K_{**} \end{bmatrix}) \tag{5}$$

K_{x*} defines the covariance between new and observed function values while K_{**} defines the covariance among new values. Same as in earlier equations these matrices are generated through pairwise applicaiton of covariance function on respective values.

Following conditional properties of Gaussian distribution [Christopher M. Bishop 2006] we can generate distribution for F* given f as:

$$P(F^* \mid F, X, X*) \approx \mathcal{N}(F^* \mid K_{*x}K_{xx}^{-1}F, K_{**} - K_{*x}K_{xx}^{-1}K_{x*}) \tag{6}$$

Finally, if we are interested only in corresponding predicted values Y* it can be computed as:

$$p(Y^* \mid Y, X, X*) \approx \int p(Y^* \mid F^*)P(F^* \mid X^*, Y, X)dF^* \tag{7}$$

Here, $p(F^* \mid X^*, Y, X)$ is the posterior predictive distribution that can be obtained by integrating out the latent function values F from equation 6

2.2.3 Hyperparameter selection

GP specifies a fully probabilistic model and hence it has a distinct advantage that hyperparameters can be inferred from training data directly instead of using a cross validation scheme as in case of other methods. In a pure Bayesian framework, hyperparameters would have priors set over them. However, complex relationship of hyperparameters with covariance function means usually the integrals become intractable and must be done through other numerical methods like MCMC or likelihood maximization. Due to computational expensiveness of MCMC, likelihood maximization is generally used for hyperparameter selection in GP. A gradient based optimization procedure can be used through Marginal data likelihood in the denominator of equation 4 to optimize hyperparameters,

$$p(Y \mid X, \theta) = \int p(Y \mid F)p(F \mid X, \theta)dF$$

Since the prior $p(F \mid X, \theta)$ is Gaussian ($\mathcal{N}(F \mid 0, K_{xx})$), after combining this with the Gaussian likelihood $p(Y \mid F) = \mathcal{N}(Y \mid F, \sigma^2)$ and integrating out F, the log marginal likelihood can be written as,

$$\log p(Y \mid X, \theta) = -\frac{1}{2}Y^T(K + \sigma^2 I)^{-1}Y - \frac{1}{2}\log(K + \sigma^2 I) - \frac{n}{2}\log \pi \tag{8}$$

In the equation 8 all the terms are a function of θ. (K is calculated by pair wise application of function $k(x_i, x_j, \theta)$ on X) and hence optimal values of hyperparameters can be obtained by finding the gradient of 8 with respect to each $\theta_j \in \theta$

Though ML estimates are sometimes susceptible to overfitting, a well peaked posterior however, should drastically reduce this risk significantly [D. J. C. MacKay 1992].

2.2.4 GP Regression

Gaussian process based Bayesian non parametric regression have been described in [Blight et al. 1975, Kendall et al. 1994, C. K. Williams et al. 1996] etc. The central idea is to first start with the parametric linear model and then generalize it through Gaussian processes. [Rasmussen and Christopher KI Williams. 2006] generalizes regression by first using basis function to project standard linear regression to a high dimensional space and then using *kernel trick* [Shawe-Taylor et al. 2004] to arrive at the same result as us in section 2.2.2, this approach is also called the weight space view of Gaussian processes, details of which can be found in [Kendall et al. 1994, C. K. Williams et al. 1996 , Seeger 2004,Rasmussen and Christopher KI Williams. 2006]. Here we take a simpler approach based on the development of previous sections. Same equations can be arrived at using weight space view as illustrated in [Williams et al. 1998] A generic regression problem can be defined as:

$$Y = F(x) + \epsilon(noise) \tag{9}$$

Here the objective is to find a good enough estimate of $F(x)$ so that one could predict Y^* values for input X^*. Traditional solution of the problem have been to use a parametric model similar to:

$$F(x|w) = w_1 + w_2 x + w_3 x^2$$

w's are also called the weights of the covariates. In Bayesian parametric setting, w's will be called the parameters and we put a prior p(w) on them. Once posterior $p(W \mid Y)$ is obtained we can predict new values using $p(Y^* \mid X^*, W)$. It can be observed that this approach quickly creates the problem of limiting the function capacity as explained in section 2.1. Moreover, it might also be difficult to encode prior beliefs about weights as a distribution $p(W)$.

However, if we consider the objective of regression task to find a suitable function out of all possible functions, an overly accommodating non linear functional space can be searched for the desired regression function by using a Gaussian process prior for $F(x)$. As explained in section 2.2 using GP also alleviate the problem of providing our prior beliefs about parameters through covariance function and Kernels and we don't need to worry about weights at all.

We derive GP based nonparametric solution for regression problem for equation 9 by first putting a GP prior on the function, i.e.,

$p(F|X) = \mathcal{N}(F|0, K_{xx})$

where K_{xx} is the covariance matrix for input X.

We further assume Gaussian distribution for noise term in equation 9:

$$\epsilon = \mathcal{N}(0, \Sigma)$$

where $\Sigma = \sigma^2 I$

Now following 9 likelihood of the observations is given by,

$$p(Y \mid F, X) = \mathcal{N}(F, \Sigma)$$

From here we can find the posterior in the similar way as in the section 2.3.2:

$$p(F \mid X, Y) \approx \mathcal{N}(\mu_f, S_f)$$

where, $\mu_f = (K_{xx}^{-1} + \Sigma^{-1})^{-1}\Sigma^{-1}Y$ and $S_f = (K_{xx}^{-1} + \Sigma^{-1})^{-1}$

Similarly, as in section 2.2.2, posterior predictive distribution for a new function value f* at point x* can be directly calculated using equation 6

Regression is one of the simplest of Gaussian process use cases. [Seeger 2004] provides pointers on to other variants of this model by considering an arbitrary noise distribution directly in GP prior or further extending it for more complex generalized linear models.

2.2.5 Classification

Classification is another common problem in machine learning in which desired functional value is categorical. In a generic classification setting, our observations include class labels $Y \sim [1, 2 \dots M]$ corresponding to each data point X and our objective is to estimate a function $F : X \to [1, 2 \dots M]$ that can assign these class labels appropriately to the new data points X*, in Bayesian context this process is equivalent to finding the probability $p(y^* = c \mid x*, Y, X)$ where $c \in [1, 2 \dots M]$ The simplest case of GP classifier is called Bayesian discriminator [Rasmussen and Christopher KI Williams. 2006, C. Williams et al. 1998] in which we consider a GP prior over the latent function and then the latent function values for data point are turned into class probabilities using a response function. This conversion of latent values of domain $\mathbb{R}$ to output of response function of domain $(0, 1)$ guarantees a valid probabilistic interpretation of our prediction. A common method to map the function values to probabilities is to use the logit function such that if $y \sim -1, 1$ and F_i is the latent function values for X_i:

$$p(y_i \mid F_i) = \sigma(y_i, F_i) = \frac{1}{(1 + exp^{(-F_i)})}$$

Since the class labels are independent from each other given latent function values, likelihood can be written as:

$$p(Y \mid F) = \prod_{i=1}^{N} p(y_i \mid F_i) = \prod_{i=1}^{N} \sigma(y_i, F_i)$$

Once again we put a GP prior on the latent function values $P(F \mid X) = \mathcal{GP}(F \mid \mu, K_{xx})$ and write the posterior as:

$$p(F \mid Y, X) = \frac{p(Y|F)p(F|X)}{p(Y \mid X)} = \frac{\mathcal{N}(F \mid 0, K_{xx})}{p(Y \mid X)} \prod_{i=1}^{N} \sigma(y_i, F_i) \tag{10}$$

Similarly, posterior predictive distribution of latent function is,

$$p(F^* \mid X^*, Y, X) = \int p(F^* \mid X^*, F, X)p(F \mid X, Y)dF^* \tag{11}$$

Here $p(F^* \mid X^*, F, X)$ is Gaussian and can be obtained through conditional GP. However, due to $p(Y|F)$ and $p(Y|X)$ being non Gaussian, both posterior and posterior predictive distribution become analytically intractable and has to be approximated. Since categorical predictions y^* are much more important in classification context, we can still proceed for y^* by integrating out the latent function values,

$$p(y^* = +1 \mid X, Y, X^*) = \int p(y* \mid F^*)p(F* \mid X, Y, X^*)dF^*$$

Once again, same as in previous case the integration becomes intractable and we must resort to approximations to obtain the distribution details. Section 2.3 describes variational inference scheme to find an approximation in the cases like this where an analytical solution can not be calculated. Apart from variational inference scheme described in next chapters, numerous other approximation solutions have been developed in for GP classification. An extensive survey of different approximation schemes for Classification using GPs can be found in Kuss et al. 2005]. A detailed treatment of more general cases of classification like multi-class classification can be found in [C. Williams et al. 1998].

2.2.6 Sparse Gaussian Processes

We might notice from equation 4 that posterior computation requires inversion of a $N \times N$ matrix $K_{NN} + s^2 I$, an $O(N^3)$ complexity tasks with $O(N^2)$ memory storage requirement. Additionally, for hyperparameter optimization, we need to invert it at every gradient step for likelihood marginalization as shown in equation 8. Moreover, even the prediction costs as can be seen from 6 scale with $O(N^2)$ for each test case. These rapidly increasing resource requirements can render Gaussian processes computationally in-feasible for all but very small to moderate datasets.

Due to these resource intensiveness of GPs, a significant amount of work in literature has been towards making GPs computationally feasible and efficient. [E. L. Snelson 2008, Jaakko Riihimäki n.d., E. Snelson et al. 2006] all contain good overviews of different approaches for GP scaling. Most of the scalable solutions for GPs revolve around reducing the training data by using only small part of it based on some selection criteria[M. Titsias 2009]. The key idea being that if data points are packed together very closely, one need not use all of them to represent the function in that area. Using a much smaller number of data points $n \ll N$ one could significantly reduce the computational resources required while at the same time capturing the functional form well enough for all practical purposes.

This subset of data points is called by many names like inducing set, partitioned set etc. based on the approximation methodology [E. L. Snelson 2008] and the resulting GP as sparse gaussian processes . Most of these approximation approaches differ from each other the strategy they employ to chose this new reduced set of points. For example, [Seeger, C. K. I. Williams, et al. 2003] used a combinatorial approach to select best subset of training points suing greedy selection. Similarly, [Herbrich et al. 2003] uses a differential entropy score for subset selection. However, as noted in [E. Snelson et al. 2006], this type of selection can interfere with hyperparameter learning. Inconsistent continuous selection and deletion of data points in the subset

based on information criterion creates rough fluctuations in the gradients, resulting in non smooth convergence during gradient optimization. They proposed to circumvent this problem by finding the active set locations and hyperparameters together in one single optimization procedure. The points thus found in active set were called were called pseudo inputs – since they were not constrained to be the subset of input data. [E. Snelson et al. 2006] further demonstrated that due to the added flexibility of moving data locations around model can explore a much richer function space than the one learnable only through training data. [Quiñonero-Candela et al. 2001] provides a unifying view of sparse approximation by proposing that all of these sparse approximation schemes effectively change the GP prior by making location of active set elements to be one of the prior assumptions aka hyper-parameters, thereby changing the interpretation of sparse GP inference from "Approximate inference from exact prior" to "Exact inference on approximate prior".

[M. Titsias 2009] however noted that aforementioned approaches do not consider the convergence to true distribution and considered the lack of distance measure between the true distribution and the one found through sparse approximation a lack of rigor in approximation procedure. Furthermore, they argued that inclusion of subset locations as hyper parameters increases the effective number of hyper-parameters to optimize which might lead to a severely over-fitted model. A variational framework was instead proposed in which lower bound of marginal likelihood is used to optimize inducing inputs as well as the hyperparameters. The procedure works by minimizing the Kullback-Leibler divergence between the true approximate GP posterior and hence guarantees that approximation would always get closer to the true distribution at each optimization step.

Under this framework, the posterior GP given in equation 4 is approximated by introducing n auxiliary inducing variables $\hat{F}$ such that $\hat{F} = \hat{F}(x)$, for $x \in \{\hat{X}\}_{i=1}^{n}$, a subset of training data.

Now a combined posterior with both actual and pseudo function values can be written as,

$$p(F, \hat{F} \mid Y) = p(F \mid \hat{F}, Y)p(\hat{F} \mid Y)$$

In cases where pseudo inputs are located close enough $\hat{F}$ can encapsulate the information about latent function form accurately, we can say that any additional data would not change our knowledge about f, i.e. f and y are conditionally independent given $\hat{F}$. Thus, $p(F \mid \hat{F}) = p(F \mid \hat{F}, Y)$ and combined posterior,

$$P(F, \hat{F} \mid Y) = p(F \mid \hat{F})p(\hat{F} \mid Y) \tag{12}$$

$p(F \mid \hat{F})$ can be obtained though Gaussian conditional rules [Christopher M. Bishop 2006], while $p(\hat{F}|F)$ is approximated by a variaitonal distribution $q(\hat{F} \mid Y)$ giving rise to the sparse GP posterior,

$$q(F, \hat{F}|Y) = p(F|\hat{F})q(\hat{F}|Y)$$

and all these auxiliary variables together can be written as,

$$p(Y, F, \hat{F}) = p(Y \mid F)p(F \mid \hat{F})p(\hat{F}) \tag{13}$$

One important point to note here is that even though location of pseudo inputs $\hat{X}$ affects conditional $p(F|\hat{F})$ as well as $p(\hat{F})$, $\hat{X}$ should be treated as variational parameter instead of model parameter since $\hat{F}$ would eventually get marginalized and never impact the true posterior $p(F|Y)$ or marginal likelihood $p(Y)$[M. Titsias 2009].

2.3 Approximate Inference

As we saw in previous section more often than not, Bayesian and specially GP posteriors end up being analytically intractable and instead of the true distribution we might have to work with a close enough approximation. In this section we first briefly introduce various approximate techniques, followed by a short primer on variational Bayes framework. Finally, the section with an overview of VB derivation steps for a sparse gaussian process.

2.3.1 Introduction

In most of the cases, the solution of a Bayesian inference problem comes down to computing a posterior and a predictive posterior[Gelman, J. B. Carlin, et al. 2014]. Moreover, as we saw in section 2.2.3, calculating marginal data likelihood can also be significantly important for model selection and hyperparameter optimization. If X is the data and z the parameter that govern its generation process, Computing posterior $p(z|x) = \frac{p(x|z)p(z)}{p(X)}$ where $p(X)$ is the data likelihood and can be obtained as $p(X) = \int p(x|z)p(z)dz$. Finally prediction on a new data point x^* can be obtained as

$$P(x^*|z) = \int p(x^*|z)p(z|x)dz$$

Unfortunately, in most of practical considerations like in GP classification case (section 2.2.5), integrals mentioned in inference equations above become intractable. Traditionally, Markov Chain Monte Carlo (MCMC) methods - part of the generic stochastic family of approximations [Christopher M. Bishop 2006], have been employed to achieve the full integration results in the Bayesian problems [B. P. Carlin et al. 1995, Gelman and Meng n.d., Radford M Neal 2001]. These techniques try to approximate the distribution by generating number of samples. Though MCMC guarantees to converge to the target distribution in the long run [Andrieu et al. 2003], typically an extremely large number of samples is required to converge to accurate results. Despite numerous advances in trying to make MCMC methods faster and efficient most of them still remain computationally prohibitive for most of the relevant models. An extensive coverage of these stochastic techniques and their properties can be found in [Gelman, J. B. Carlin, et al. 2014], similarly [Andrieu et al. 2003] is a comprehensive introduction on MCMC applications from machine learning perspective.

A number of other methods called deterministic methods approach the same task by trying to approximate posterior distribution using analytical approximations [Gelman, J. B. Carlin, et al. 2014, Thomas Peter Minka 2001]. These include Maximum likelihood and maximum a posteriori methods which approximate entire distribution to corresponding single point estimates [Murphy 2012,Gelman, J. B. Carlin, et al. 2014], thereby discarding all the uncertainties information associated with it. Laplace's method, on the other hand, approximates true distribution by a Gaussian distribution that matches the mode, first derivative and second derivative of the former [D. J. C. MacKay 2003]. Another family of methods include Expectation propagation [Thomas P. Minka 2001] and Variational Bayes [Andrieu

et al. 2003] that try to obtain a simpler distribution closest to the target distribution based on information theoretic distance metrics like KL divergence. Compared to MCMC methods, deterministic methods are generally not able to recover the target distribution exactly. However their computational efficiency and versatility makes them a very useful method in approximate inference toolbox.

2.3.2 Variational-Bayes framework

Fundamental idea behind Variational Bayes framework [Andrieu et al. 2003, Beal 2003] is to approximate an intractable or in-feasible posterior distribution by a simpler distribution through optimization. The quantification of closeness between two distributions is usually done by the Kullback-Leibler divergence [D. J. C. MacKay 2003], this can serve as a good optimizer during the approximation procedure. Optimal values of the approximate distribution's parameters, also called variational parameters, are thus obtained by minimizing the KL divergence between the former and true distribution.

Following [Blei et al. 2017], If q(θ) approximates certain posterior $p(\theta|x)$, KL divergence between them can be given by,

$$\begin{aligned} KL(q||p) = \int q(z) \log \frac{q(z)}{p(z|x)} dz &= \int q(z) \log q(z) dz - \int q(z) \log p(z \mid x) dz \\ &= \log p(x) - (\int q(z) \log p(x, z) dz - \int q(z) \log q(z) dz) \end{aligned}$$

Since $\log p(x)$ is the evidence likelihood and constant with respect to z, we can see that KL divergence is minimized with maximization of rest of the term in the above equation's RHS. Same equation can be flipped to give us,

$$\log p(x) = KL(q||p) + \int q(z) \log p(x, z) dz - \int q(z) \log q(z) dz \tag{14}$$

Here, $\log p(x)$ represents the marginal likelihood of the evidence itself and since $KL(q||p) \geq 0$, second term lower bounds the evidence and called *Evidence Lower Bound* or ELBO. Furthermore, as mentioned earlier, eqn. ELBO is equivalent to KL divergence upto a constant and hence maximizing former is equivalent to minimizing the KL divergence. [Jordan et al. 1999,Christopher M. Bishop 2006] achieve the same result using Jensen's inequality.

However, supposedly intractable distribution p(z|x) still appears in ELBO optimization equation rendering the optimization procedure still intractable. The solution can be made tractable by constraining the range of functions over which optimization can be performed. In mean field theory [Peterson et al. 1987, Parisi 1998] suggests to factorize distribution $q(\theta)$ with respect to all of its parameters, however other factorization possibilities exist and usually can be specific to the model being studied [Jaakkola et al. 1997, Girolami 2001, Christopher M Bishop 1999]. If z had M factors $[z_1, z_2, ..., z_m]$,

$$q(z) = \prod_{i=1}^{M} q(z_i) \tag{15}$$

Corresponding ELBO equation can be written as,

$$\log p(x) = KL(q||p) + \int .. \int \prod_{i=1}^{M} q(z_i) \log p(x,z) dz_1 dz_2 .. dz_m - \sum_{i=1}^{M} \int \log q(z_i) dz_i \quad (16)$$

Since each z_m supposed to be independent from rest of the factors, likelihood with respect to m^{th} factor would be,

$$L(z_m) = \int q_m(z_m)(\log \hat{p}(x,z)) - \log q_m(z_m) dz_m \quad (17)$$

where,

$$\log \hat{p}(x,z) = \int .. \int \prod_{i=1}^{Mm} q(z_i) \log p(x,z) dz_1, dz_2 ... dz_m$$

Equation 16 can also be seen as equivalent to reducing KL divergence between $q_m(z_m)$ and $\hat{p}(x|z)$ [Luttinen 2009a]. Since iteration of one factor's distribution is dependent on others, it results in an iterative algorithm like EM [Dempster et al. 1977]. Details presented here are a sketch of VB-EM algorithm presented in [Beal 2003,Attias 2000]. It can be further proven that each update of the factors under this framework is guaranteed to move variational solution to a local optima [Beal 2003 p. 55-57].

2.3.3 Variational Infernece in sparse GP

We demonstrate variational-Bayes framework by approximating the sparse gaussian process posterior mentioned earlier in section 2.2.6.

As explained in previous section variational parameters like $\hat{X}$ can be determined by minimizing the KL distance between true and approximate distribution which are $q(F, \hat{F})$ and $p(F, \hat{F})$ respectively in the case of sparse GPs. Moreover, we observed in previous section that same minimization can also be achieved by maximizing the ELBO. Following [M. Titsias 2009],

$$\begin{aligned} L(Q, \theta, \hat{X}) &= \int \int p(F|\hat{F}) q(\hat{F}) \log \frac{p(Y, F, \hat{F})}{p(F|\hat{F}) q(\hat{F})} dF d\hat{F} \\ &= \int q(\hat{F}) \log \frac{\hat{p}(Y|\hat{F}) p(\hat{F})}{q(\hat{F})} d\hat{F} \end{aligned}$$

where $\log \hat{p}(F|\hat{F}) = \int p(F|\hat{F}) \log p(Y|F) dF$

Second last equation above can also be interpreted as the KL divergence between $q(\hat{F})$ and $p(Y|\hat{F})p(\hat{F})$ which is also similar to the posterior distribution of a model with likelihood $p(Y|\hat{F})$ and $p(\hat{F})$ as the prior.Following [Luttinen 2009a] $q(\hat{F})$ can be obtained as,

$$q(\hat{F}) \approx \mathcal{N}(\hat{F}|S_f K_{nn}^{-1} K_{nN} \Sigma^{-1} Y, S_f) \quad (18)$$

where $S_f = (K_{nn} + K_{nN}^{-1}\Sigma^{-1}K_{Nn}K_{nn}^{-1})^{-1}$ and $\Sigma = \sigma^2 I$ is the noise term corresponding to GP prior.

As can be seen from equation 18 though very similar to GP posterior, the original space is reduced by a factor of $\frac{n}{N}$ due to the projection onto a lower dimensional space of auxiliary variables making sparse approximation is much more computationally feasible to calculate.

Once the optimal distribution $q(\hat{F})$ is found, θ and locations of inducing variables $\hat{X}$ can be optimized through ELBO maximization. Following [M. K. Titsias 2009] The lower bound to be maximized is obtained as:

$$L(\theta, \hat{X}) = \log \mathcal{N}(Y|0, \Sigma + K_{Nn}K_{nn}^{-1}K_{nN}) - \frac{1}{2}Cov(F|\hat{F})\Sigma^{-1} \tag{19}$$

[M. Titsias 2009] notes that first term in the equation is the projected process proposed before by [Seeger, C. K. I. Williams, et al. 2003] that encourages approximation to fit well to the data. The latter term however is additional regularizer that represents the total variance of conditional $p(F|\hat{F})$. This trace term can be interpreted as the 'gap in knowledge' occurred due to usage of inducing inputs instead of original data points. The regulaizer term thus, tries to maintain the difference between approximation and true posterior close to each other by working against the rejection of locations in inducing set that might contain more information than others points about the true function.

From an implementation perspective, since ELBO optimization does not depend on the exact form of $q(\theta)$ but equation 18 depends on the values of hyperparameters, we can either optimize hyperparameters first and use their values in variational algorithm or can alternate between optimization and variational updates.

In cases of high dimensional input spaces, gradient based optimization to selected inducing inputs can become difficult. [M. Titsias 2009] provides a way to select pseudo inputs as the subset of the data by applying a greedy selection scheme based on an EM like algorithm. This variational EM with interleaving of hyperparameter optimization including selection of inducing set guarantees the monotonic increase of lower bound [M. K. Titsias 2009].

3 Latent Variable Models

Latent variable models have been a cornerstone of complex, high dimensional data analysis for a long time. They are slightly different from the methods described earlier in the sense that with LVM one tries to understand the underlying data structure by assuming that the observations are a result of interaction between an unknown number of hidden or latent variables. The objective of latent variable models, thus becomes to learn these hidden variables along with their interaction process.

The latent representations learnt during data analysis can be very task specific. For example, these representations can summarize the most relevant aspects of high dimensional data in fewer dimensions in order to reduce data-dimensionality or can aid in recognizing patterns by extracting important non measurable aspects or features from data. Most importantly, latent variable models provide a simple flexible framework under which various different statistical methods can be unified [Bartholomew et al. 2011].

If we consider latent variable as the parameters of data, the idea of learning hidden variables and the process through which they generate observations closely follow the Bayesian inference process described in section 2. Additionally, inverse of this data generation process can also be referred as projection of data to latent space. The interaction among latent variables or mapping to latent space can be linear or nonlinear. Linear interactions like FA, PCA [Jolliffe 2002], ICA [Hyvärinen et al. 2004] etc, though interpretable due to their simplicity and computationally fast, may not be suitable in cases where a complex latent space mapping is required. Nonlinear interactions on the other hand can provide enough capacity to model complicated processes at the cost of interpretability and computational ease. Self organizing maps [Kohonen 1998], nonlinear PCA [Scholz et al. 2008], Gaussian process latent vector model [Neil D Lawrence 2004] etc are some of the important nonlinear latent variable models. [Raiko et al. 2007] provides a structured approach to build various types LVMs.

In the following subsections we first describe the basic linear model called Factor Analysis from a probabilistic perspective and later use gaussian processes to enhance the FA model in section 3.2 closely following the semiparametric latent factor modelSeeger, Teh, et al. 2004. In section 3.3.we further extend the resulting model to handle multi-observational settings and conclude this chapter with a short discussion about our extensions.

3.1 Probabilistic Factor Analysis

Originated from psychological research in early 19th century, Factor models are considered one of the oldest linear latent variable methods. Their name is derived from two factor model presented by [Spearman 1904] postulating that test scores of individuals can be thought of as linear combinations of two factors, namely common factor- akin to one's general intelligence and another specific to the context of test [Anderson 1956].

Central objective behind factor analysis is to describe the co-variance among

large number of observed variables in terms of less number of underlying latent variables, called factors. This ability to explain away the variation can also enable factors to express commonalities hidden in data, making them very suitable tool for extrapolatory data analysis.

Consider an observed data set $Y \in \mathbb{R}^{N\times P}$ and factors $X \in \mathbb{R}^{N\times Q}$, where commonly $Q \ll P$. According to traditional FA model[Anderson 1956,Bartholomew et al. 2011], each instance of Y is assumed to be generated by a linear transformation of uncorrelated factors X such that ,

$$Y_i = \Phi X_i + \mu + E_i, \tag{20}$$

Where, Φ is a constant matrix representing the linear transformation of factors X. Values of Φ correspond the amount of influence a particular factor has on the observed data and thus are appropriately named as factor loadings, μ is the mean vector. Factors X_i as well as noise E_i are assumed to be Gaussian distributed as $\mathcal{N}(X_i|0, I)$ and $\mathcal{N}(E_i|0, \Psi)$ respectively.

This gives us a normally distributed generative model of data given by,

$$Y_i \propto \mathcal{N}(Y_i|\mu, \Phi\Phi^T + \Psi) \tag{21}$$

It can be observed that if Ψ is a diagonal matrix estimated from observations, covariance in Y can be solely explained by the latent factors while E_i models the individual variability of each data instance Y_i. A detailed review of various properties of FA models can be found in [Bartholomew et al. 2011].

Due to its popularity, a number of inference methods both Bayesian and non-Bayesian have been proposed to estimate FA parameters over time [Anderson 1956,Lawley 1940,Rubin et al. 1982,Rowe et al. 1998,Press et al. 1989,Jöreskog 1967]. In addition to the ability of including prior or expert knowledge about factors, Bayesian approaches have added consequence of eliminating rotational ambiguity in estimation of factor loadings.

In Bayesian formulation of Factor analysis, we put a prior over factor loading matrix Φ and substitute error covariance matrix Ψ with its inverse, precision matrix ψ.

$$\begin{aligned} p(\Phi) &= \prod_{q=1}^{Q} \mathcal{N}(\phi_q|0, f_q^{-1} I) \\ p(f) &= \prod_{q=1}^{Q} \mathcal{G}(f_q|a_q^f, a_q^f) \\ p(\psi) &= \mathcal{G}(\psi|a^\psi, b^\psi) \quad , \psi = \Psi^{-1}. \end{aligned}$$

Here, ϕ_k represents kth column of the loading matrix (or kth factor) which is isotropically Gaussian distributed with the precision matrix represented by f_k.Precision matrices ψ and f_k are given Gamma priors controlled by their respective hyper-parameters. Being the precision of zero mean Gaussian, f_k can be used to shut off corresponding redundant factors.

Joint posterior distribution of parameters $p(\Phi, X, \mu, \psi, f|Y)$ is analytically intractable and must be approximated. [Nielsen 2004,Ghahramani and Beal 2000] provide detailed derivations for variational inference over a fully factorized approximated distributions of this posterior. However, these methods are known to have some downsides like heavier penalization of model or inability of solution to shut down redundant factors specially in observations with minimal noise which was more recently overcome by [Zhao et al. 2009] in their improved VB schemes.

Distributional assumptions of factors and noise terms in FA are of utmost importance since slight variations in them result in a variety of different models with significantly different characteristics. Principal component analysis [Jolliffe 2002], for example is considered a special case of FA in which noise is kept same for every factor (i.e. $\Psi_k = \Psi$). [Tipping et al. 1999] further showed that as Ψ approaches infinity ML estimation of FA tends to approach standard PCA results. Similarly, a non Gaussian assumption of the factors lead us to ICA model, enabling factors (or components) to be truly independent from each other. Introduction of fixed or time varying state dynamics within factors is called linear state space model, another extension of FA models [Luttinen et al. 2014]. All of these models are closely related to each other due to their linear Gaussian assumptions. A detailed review of latent linear Gaussian models can be found in [Roweis et al. 1999].

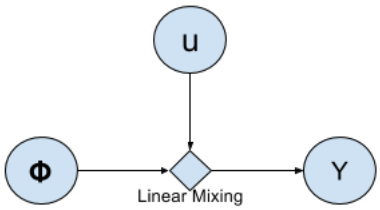

Figure 3: Graphical representation of GP based factor analysis model

3.2 GP based Factor analysis model

One way to achieve temporal structures within factor models is by putting GP priors over the factors. This lets us specify the relationship between latent values of a factor with the help of covariance functions.

These methods are popular in scenarios where observations correspond to some varying phenomenon like time series [Yu et al. 2008] and/or locations [Luttinen 2009b]. Moreover, GP based LVMs have also been used in a multiple response setting [Seeger, Teh, et al. 2004] where GP were used to model the dependencies between response variables. These are called semi-parametric models due to them being a combination of Non-parametric properties of GP and parametric linear mixing of factors.

[Seeger, Teh, et al. 2004] proposed a very similar model in a multiple response setting and uses IVM framework [Herbrich et al. 2003] for the posterior approximation. However, computation with multiple GPs can be hefty with larger data set and in this thesis we aim to provide a VB based inference along with a sparse approximation that should be computationally efficient even in cases of larger data sets.

Mathematically, model is very similar to the standard factor model described in previous section with the distinction that here latent factors U are assumed to be conditionally independent GP, indexed through a common index x. As can be observed from Figure 3, flow of information from each observed response to

every latent factor response makes the sharing of statistical strength among response variables possible, thereby resulting in a much more expressive model than traditional factor analysis models. The model is shown in figure 3 and can be expressed as,

$$Y = \Phi U + \sigma^2 I \tag{22}$$

where $u \in \mathbb{R}^{P \times N}$ are the latent factors that get linearly mixed through matrix $\Phi \in \mathbb{R}^{C \times P}$, resulting in the observations $Y \in \mathbb{R}^{C \times N}$. As in [Seeger, Teh, et al. 2004], factor analysis emerges when $P << C$

We put a GP prior on latent variables $u_p \sim GP(0, K^p)$ where K^p is the co-variance kernel for that particular Gaussian process. Loading matrix Φ is assumed to have a Gaussian prior $\mathcal{N}(0, I)$.

3.2.1 Variational Sparse Approximation of GPFA posterior

To achieve sparsity in posterior, we approximate the posterior distribution as $q(\phi, u, \hat{u})$ where $p(\hat{u})$ is the distribution through n inducing points (n « N). Finally, a GP prior is put on $p(\hat{u})$ too and the sparse approximation of posterior to factorizes as:

$$q(\phi, u, \hat{u}) = q(\phi) \prod_{p=1}^{P} p(u_p \mid \hat{u_p}) q(u_p) \tag{23}$$

Based on above sparse approximation, marginal likelihood of data can be given as,

$$L(Y|X) = \int p(Y|\hat{u}, \phi) p(\phi, \hat{u}, u) d\phi du d\hat{u} \tag{24}$$

Following the VB inference scheme as outlined in section 3, we find optimal distributions by minimizing the KL divergence between approximation and true posterior. This in turn is equivalent to maximizing the log of marginal likelihood given in equation 24 with respect to each parameter(detailed derivations are given in Appendix 1) :

For $\hat{u_p}$,

$$q(\hat{u_p}) \propto \mathcal{N}(\hat{u_p} \mid \Sigma_p^{-1} K_{nn}^{-1} K_{Nn} Z_p, \Sigma_p^{-1})$$

where, $\Sigma_p = K_{nn}^{-1} + \frac{1}{\sigma^2} K_{nn}^{-1} K_{nN} S_p K_{Nn} K_{nn}^{-1}$

and $Z_p = \sum_c^C \mathbb{E}[\Phi_{cp}](y_c - \sum_i^{P/p} \mathbb{E}[\Phi_{ci}]\mathbb{E}[u_{ip}])$.

Similarly VB update for $\hat{\phi}$ can be derived as,

$$q(\Phi) \approx \mathcal{N}(\Phi \mid y\mathbb{E}[U]^T \Sigma_\phi^{-1}, \Sigma_\phi)$$

where $\Sigma_\phi = (V_\phi^{-1} + I)^{-1}$ and $V_\phi = \mathbb{E}[U]\mathbb{E}[U]^T \sigma^2$

Finally update for u_p is,

$$q(u_p) \approx \mathcal{N}(u_p \mid M\hat{\mu_p}, \Sigma_{u|\hat{u}}^p + M\Sigma_p M^T)$$

where $\hat{\mu_p}$ is the mean of GP u_p, $M = K_{Nn} K_{nn}^{-1}$ and $\Sigma_{u|\hat{u}}^p = K_{NN} - MK_{nN}$

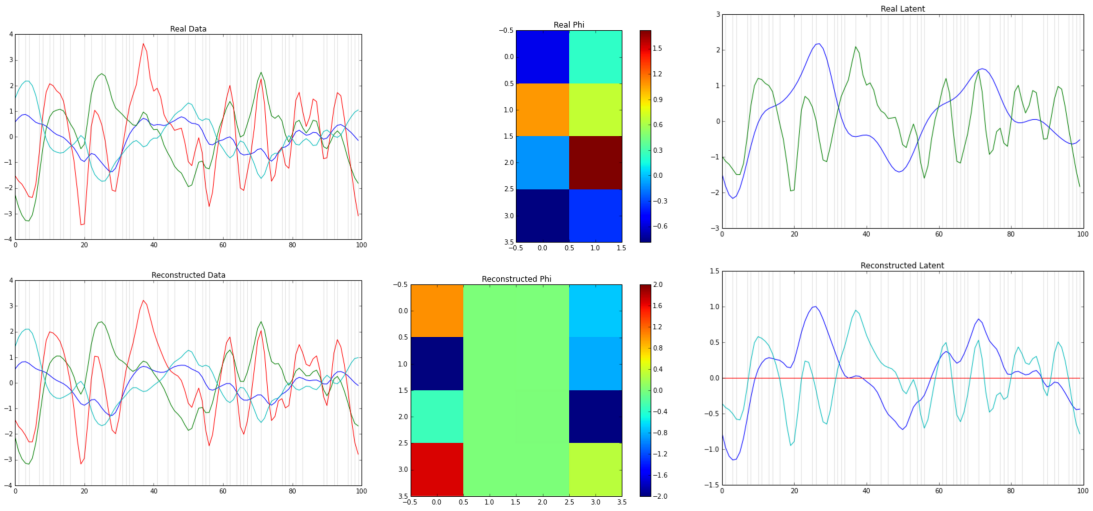

Figure 4: Extracting latent processes from artificial data set using GPFA: Upper row shows the real latent process and mixing matrix. Second row demonstrates the extracted latent structure using variational inference. Gray Spikes indicate the inducing points at which functions were evaluated

3.2.2 Hyper-parameter selection

Hyperparameters for GP based FA that might require tuning mostly include the parameters for covariance kernel function of GPs corresponding to the latent factors. These can be optimized by maximizing the marginal log likelihood with respect to the latent factors as:

$$L(\theta_p) = log\mathcal{N}(S_p^{-1} z_p \mid 0, S_p^{-1} + K_{Nn} K_{nn}^{-1} K_{nN}) - \frac{1}{2} tr(\sum_{i=1}^{P} S_i * cov(u_i \mid \hat{u}_i))) + const. \tag{25}$$

where $S_p = \sum_c^C \mathbb{E}[\phi_{cp}^2]$

Using above equation one can obtain the gradients by differentiating the log likelihood corresponding to each parameter θ_p of co-variance kernels associated with respective late factor GP. Inference then could be performed wither by first optimizing hyperparameters and then VB updates while keeping the hyperparameters fixed or iterating between hyperparameter optimization and VB updates.

3.2.3 Demonstrations on artificial data set

In order to demonstrate model's ability to recover latent processes that might have been used in data generation processes, We generated two artificial data sets using distinct latent process sets.

First data set was created by mixing two latent processes (using 100 evaluation points) with the squared exponential kernel and using a randomly generated mixing matrix. We then initialized the variational algorithm with four random latent processes with SE of lengthscale and variance one. The inducing set u_p was chosen to be only half of the original data set selected randomly. As can be seen in the

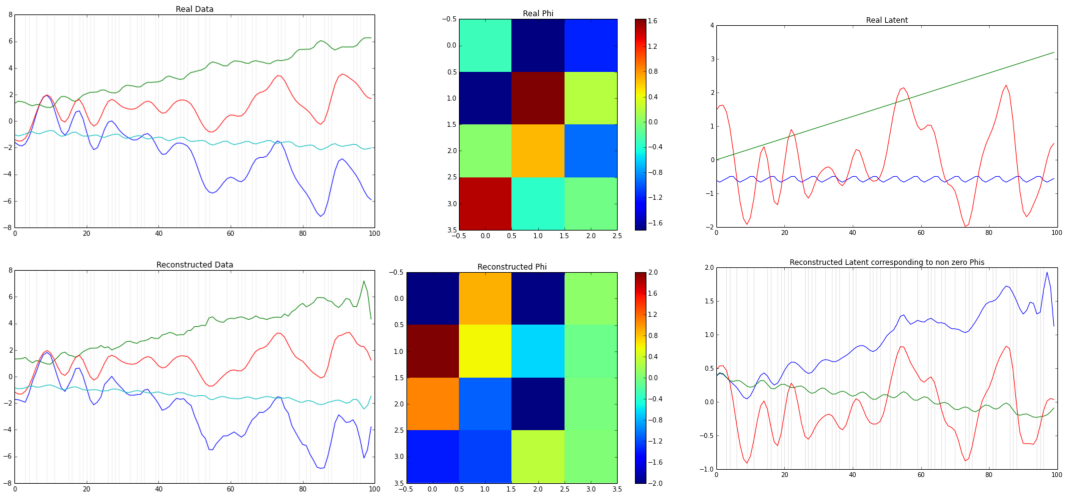

Figure 5: Extracting latent processes from artificial data set using GPFA: Upper row shows the real latent process (with periodic, linear and Gaussian kernels) and mixing matrix. Second row demonstrates the extracted latent structure using variational inference. Gray Spikes indicate the inducing points at which functions were evaluated

Figure 4, the model was not only able to recover the latent process structure very well but also shut off two redundant latent processes acting very similar to ARD prior [D. J. MacKay 1994]. Since the variational approximation recovers complete distribution of the latent processes as well as the loading matrix, it is possible to augment these values with the confidence (something that's not possible in SLFM for mixing matrix).

Since in most practical scenarios the latent process characteristics are unknown, an unsupervised model should be able to produce output that would be ultimately valuable to the user, i.e. would inform the user about characteristics of the dataset being analyzed. In this example, we generated data by mixing three different latent processes with the periodic, linear and Gaussian kernel using a randomly generated mixing matrix. To mimic the ignorance of analyst about data generating process we once again used four random Gaussian processes with only SE Kernels to initialize the inference algorithm. As can be seen from the results shown in Figure 5, though imperfect, model is able to extract the cyclic and linear structures inherent in latent processes. Moreover, as in the previous example, model can automatically determine the redundancy and is able to shut off the extra latent process that we proposed in the beginning.

3.3 Extending GP based factor analysis

Apart from the processes varying in time one of the main requirements in factor analysis is to support instance based scenarios, i.e. multiple instances with each instance having multiple response variables indexed by time. EEG data set [Tolvanen et al. 2014]is one of the prime example of the situation where each instance corresponds

to a subject and variables include 50 second long readings of multiple sensors. The central motivation is to model scenarios where each instance can have its own specific inherent processes. Figure describes the model, now the output or observed data is considered to be in $S \times C \times N$ while U in $S \times P \times N$ in where S represents the instances or trials and P latent factors. To accommodate multiple trials we fold Y and u in vectorized form i.e, $Y \in \mathbb{R}^{S \times CN}, U \in \mathbb{R}^{S \times PN}$ where $U \in GP(0, \bar{K})$ and $\bar{K}$ is a block diagonal matrix of size PN x PN with covariance kernels $K_p \in R^{N \times N}$ for each latent GP on it's diagonals, giving us follwoing generative model,

$$Y = U\bar{\Phi}^T + \sigma^2 I \tag{26}$$

$\bar{\Phi} = \Phi \times I_{N \otimes N}$ is also a block diagonal of size $CN \times PN$ with $C \times P$ matrix ϕ on its diagonal.

We apply the same procedure as in section 3.4 to find the posterior

3.3.1 Approximations for Extended GPBFA

Following the same procedure and notation as for the GP based factor analysis the sparse variational approximation for the extended format of GPFA can be given as,

For sparse distribution $\hat{u}$,

$$q(\hat{u}_p) \propto N(\hat{u}_p \mid y\mathbb{E}[\phi] K_{pNpn} K_{pnpn}^{-1} \hat{\Sigma}_u^{-1}, \hat{\Sigma}_u^{-1})$$

where

$$\hat{\Sigma}_u = K_{pnpn}^{-1} + \frac{1}{\sigma^2} K_{pnpn}^{-1} K_{pnpN} F_u K_{pNpn} K_{pn}^{-1},$$
$$F_u = \mathbb{E}[\bar{\phi}^T \bar{\phi}] = Var(\bar{\phi}) + \mathbb{E}[\bar{\phi}]^T \mathbb{E}[\bar{\phi}]$$

Similar to earlier section u can be obtained as,

$$q(u_p) \propto N(\hat{\mu}_p M, \Sigma_{u_p|u^p} + M \Sigma_u M^T)$$

where $\hat{\mu}_u$ is the mean and $\hat{\Sigma}_u$ variance of $\hat{u}$, $\Sigma_{u|u^p} = K_{pNpN} - K_{pNpn} K_{pnpn}^{-1} K_{pnpN}$ and $M = K_{pnpn}^{-1} K_{pNpn}$.

ϕ can be obtained as,

$$\phi = N(\phi | (\bar{V}_\phi + I)^{-1} \bar{z}_\phi, (\bar{V}_\phi + I)^{-1})$$

where $\bar{V}_\phi = \sum_s^S (\mathbb{E}[\bar{u}_s] \mathbb{E}\bar{u}_s]^T + \mathrm{I})$ and $\bar{z}_\phi = \sum_s^S \mathbb{E}[\bar{u}_s] \bar{y}_s^T$ here $x = vec(\bar{x})$.

3.3.2 Demonstration of extended GPBFA

To demonstrate the ability of extended GPBFA to extract different latent processes inherent in each instance we use an artificially generated dataset in which each instant is obtained by mixing two GPs (Figure 6). For identification purposes we make one latent GP to have significantly shorter length scale than the other. GPs for inference algorithm are initialized randomly with constant lengthscale and random mixing matrix. Figure 7 shows the correctly inferred binary mixing matrix along with redundant processes shut off. We also display comparisons between the actual and inferred latent processes of two random instances in Figure 7. Please note that all the results were obtained by using only 80% of the data points.

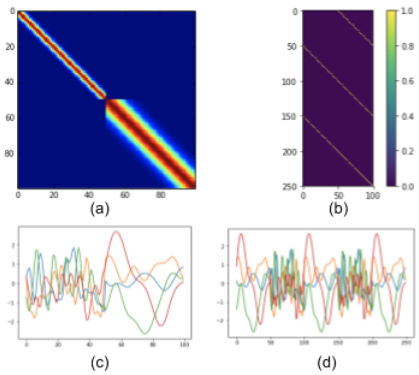

Figure 6: Artificial data set generated for Extended GPBFA: Fig(a): SE Kernel for latent GP (with shorter and longer length scales), Fig(b): mixing matrix. Fig(c): actual latent GPs of few samples, Fig(d): resulting data generated by mixing latent processes of Fig(c)

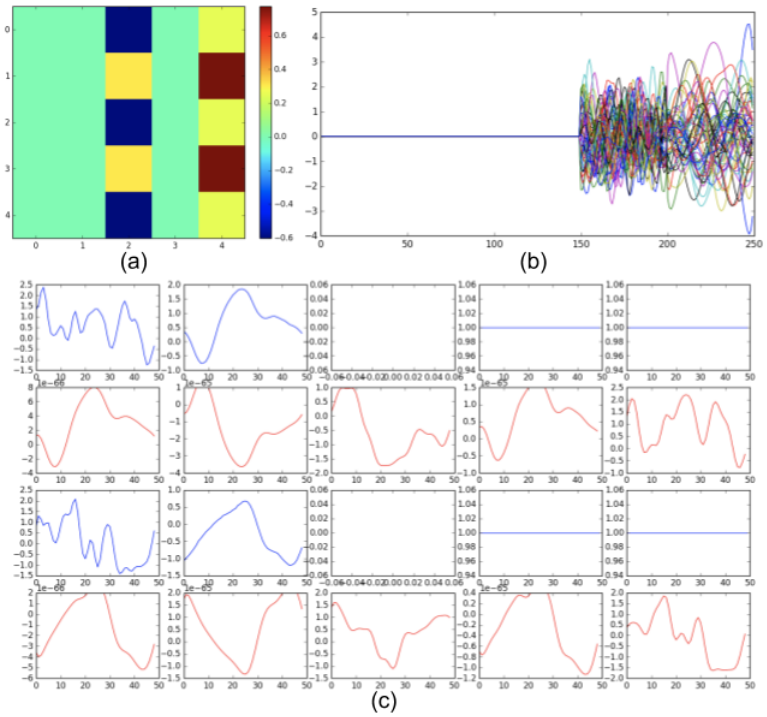

Figure 7: Extracting latent processes using extended GPBFA: Fig (a): Extracted mixing matrix,Fig (b):extracted latent processes for dataset (difference in lengthscales of extracted processes is pretty evident), Fig (c): Comparison between actual latent process (blue) and extracted ones (red) for few samples

3.4 Conclusions

In this section we first reviewed probabilistic linear factor analysis models. The model later described can be thought of as an extension to [Seeger, Teh, et al. 2004], they use IVM framework for inference of latent processes and provide point estimate for mixing matrix. In our treatment of GPBFA we derived a sparse variational approximation for latent processes which provide complete probability distributions for both latent processes as well as the mixing matrix. Moreover, our experiments with artificial data showed that it our sparse solution is able to recover latent structure by using only half of the actual data thereby making the proposed inference process more scalable and time efficient.We further extended the GPBFA model to include multi-trial/instance support and demonstrated that a sparse solution is still capable of recovering separate latent processes from each instance effectively.

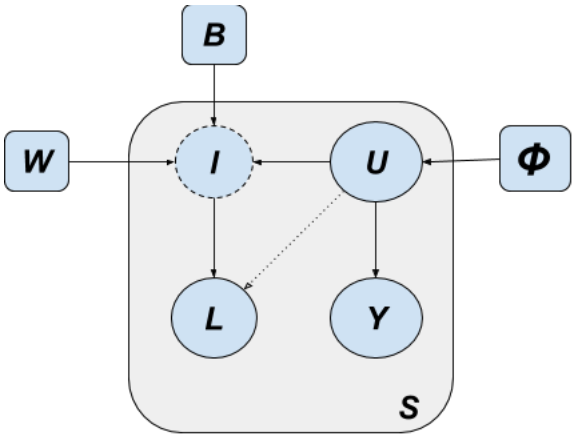

Figure 8: Graphical representation of Gaussian Process based latent classifier (lgpc)

4 Latent Gaussian Process Classifier

Expanding on the same idea of separate latent GPs for multiple sequences of each trial, in this section we introduce a novel model that combines logistic classifier with our extended GPBFA model. This GP based latent classifier for high dimensional temporal instances proves to be an effective classifier since it not only projects the observations into latent space but also learns a discriminatory plane in that space simultaneously.

4.1 The Model

In previous sections we studied models that are able to learn the inherent structure from high dimensional instances of observations, in this model we combine the latent model with a classifier and use a variatioanl algorithm to get closer to the optimal distributions one step at a time.

Consider observations Y, L, where $Y \in \mathbb{R}^{S\times C\times N}$ and each observational instance $Y[s]$ is associated with a label $\{L, L\} \in -1, 1^{S\times 1}$ signifying it to be in one class or another.

Conceptually one can imagine that both data and its labels arise from the same latent space. The discriminatory information about the samples (labels) thus, can be further utilized to learn the most relevant latent space, i.e. the one in which data can be separated best.

As can be seen from figure 8, we introduced a dummy variable l to help us keep track of classifier values in the latent space. For each data sample $s \in S$, a set of latent processes $u \in U, u \in \mathbb{R}^{P\times N}$ are assumed to generate observed sequence $Y[s] \in \mathbb{R}^{P\times N}$ through mixing matrix $\Phi \in \mathbb{R}^{C\times P}$ and a class label $L[s] \in [-1, +1]$. To learn the separation plane in latent space we attach a logistic classifier between U and L such that in additional to Y, U also generates lR^1. Parameters of classifier are $W \in R^S$ and a bias term B. Values of l then can be converted into class labels through a static rule $L[s] = \{1 \text{ if } l[s] > 0, -1 \text{ otherwise }\}$.

Similar to the model in previous section we vectorize latent and actual observations for computation purposes, making $U \in \mathbb{R}^{S \times CN}, U \in \mathbb{R}^{S \times PN}$, giving the generative model as,

$$Y = U * \bar{\Phi}^T + \sigma^2 I \tag{27}$$

Once again $\bar{\Phi} = \Phi \otimes I_{N \times N}$ is also a block diagonal of size $CN \times PN$ with $C \times P$ matrix Φ on its diagonal. Corresponding latent label values l are generated as,

$$l = B + U * W^T + s^2 I \tag{28}$$

and finally the Labels for each data instance can be generated through the rule $L[s] = -1$ if $l[s] < \mu L[s] = +1$, otherwise Thus combined data generation model along with priors can be given as,

$$p(W) \sim \mathcal{N}(W \mid 0, I) \tag{29}$$

$$p(B) \sim \mathcal{N}(B \mid 0, I) \tag{30}$$

$$p(\Phi) \sim \mathcal{N}(\Phi \mid 0, I) \tag{31}$$

$$p(l|U, B, W) \sim \mathcal{N}(l \mid W^T U + B, s^2 I) \tag{32}$$

$$p(L|l) \sim \delta(Ll > \nu); \nu = 0 \tag{33}$$

Additionally, We put a GP prior on latent variable $u \in GP(0, \bar{K^p})$ associated with each observational instance where $\bar{K}$ is also a block diagonal of size $PN \times PN$ with co-variance kernels $K_p \in R^{N \times N}$ for each Gaussian process on diagonals.

Also, just like in previous model we introduce inducing variable $\hat{u} \in \mathcal{R}^{P \times n}$ for latent processes such that $(n \ll N)$ to make the GP inference scalabale. $\hat{u}$ has a GP prior as in previous sections.

We use a variational inference scheme to approximate the posterior $p(W, B, \hat{u}, l \mid Y, L)$.

4.2 Sparse variatioanal approximation in LGPC

Following inference as sketched in previous sections we introduce variational distribution $q(W, B, l, \hat{u})$ to approximate the posterior for the model parameters. Variational updates for these parameters can be given as,

$$q(\hat{u}) \approx \mathbb{N}(\hat{u} \mid (y\mathbb{E}[\phi] + \mathbb{E}[w]\mathbb{E}[l]\sigma^2 - \mathbb{E}[w]\mathbb{E}[b]\sigma^2) K_{pNpn} K_{pnpn}^{-1} \Sigma_u^{-1}, \hat{\Sigma_u^{-1}}) \tag{34}$$

where

$$\hat{\Sigma_u} = K_{pnpn}^{-1} + \frac{1}{\sigma^2} K_{pnpn}^{-1} K_{pnpN} F_u K_{pNpn} K_{pnpn}^{-1}$$

$$F_u = \mathbb{E}[\phi^T \phi] + \sigma^2 \mathbb{E}[w^T w] = Var(\phi) + \mathbb{E}[\phi]^T \mathbb{E}[\phi] + Var(w) + \mathbb{E}[w]^T \mathbb{E}[w]$$

Using conditional and affine properties of GP we obtain real latent processes as,

$$q(u) \approx \mathcal{N}(\hat{\mu}M, \Sigma_{u|u^p} + M\Sigma_u M^T) \tag{35}$$

where $\hat{\mu}_u$ is the mean and $\hat{\Sigma}_u$ variance of $\hat{u}$
Also, $\Sigma_{u|u^p} = K_{pNpN} - K_{pNpn}K_{pn}^{-1}K_{pnpN}$ and $M = K_{pnpn}^{-1}K_{pNpn}$
Mixing matrix,

$$q(\phi) \approx \mathbb{N}(\phi \mid (\bar{V}_\phi + I)^{-1}\bar{z}_\phi, (\bar{V}_\phi + I)^{-1}) \tag{36}$$

where $\bar{V}_\phi = \sum_s^S(\mathbb{E}[\bar{u}_s]\mathbb{E}[\bar{u}_s]^T + \mathrm{I})$ and $\bar{z}_\phi = \sum_s^S \mathbb{E}[\bar{u}_s]\bar{y}_s^T$ such that $x = vec(\bar{x})$
Label values in the latent space,

$$q(l_i) \approx \mathcal{TN}(l_i \mid (\mathbb{E}[w]\mathbb{E}[u] + \mathbb{E}[b], 2I, l_i < \mu) \tag{37}$$

Here $\mathcal{TN}$ is the truncated normal distribution and μ the cut off, fixed at 0.
Parameter weights for classifier,

$$q(W) \approx \mathbb{N}(w \mid [F_u + I]^{-1}(\mathbb{E}[u]^T\mathbb{E}[l] - \mathbb{E}[u]^T\mathbb{E}[B]^T, [F_u + I]^{-1}) \tag{38}$$

and $F_u = \mathbb{E}[U^T U]$
Finally the bias term,

$$q(B) \approx \mathbb{N}(B|(\mathbb{E}[l]^T - \mathbb{E}[U]\mathbb{E}[W]^T, I) \tag{39}$$

Derivation details of these approximations follow the same procedure as GP based factor analysis given in Appendix A.

4.3 Prediction using LGPC:

Category prediction for a new observation follows standard GP prediction steps. For a new data observation Y*, first we project it to the latent space using 34.l* in the latent space can be found using predetermined values of weight parameter and bias. Finally, the label L* can be predicted using static rule for $p(L * |l*)$ mentioned in 33. However, since we do not know the values in latent space in advance we have to drop all the terms dependent on them form the expressions.

$$q(u^*) \approx \mathcal{N}(u^*|(Y^*\mathbb{E}[\phi])K_{pNpn}K_{pnpn}^{-1}\Sigma_u^{-1}M, M\Sigma_u M^T)$$
$$q(l^*) \approx \mathcal{TN}(l^*|\mathbb{E}[W]\mathbb{E}[U^*]^T + B)$$

This also means that some information is lost during prediction and hence it might not be as efficient.

4.4 Demonstration using LGPC

For demonstration purposes we use an artificial dataset in which our positive classes are genrated through a special latent process that's a combination of linear and gaussian kernels. This gives us a sort of linear trend. Latent processes for the negative classes are generated using simple gaussian kernels. We pass these through a randomly

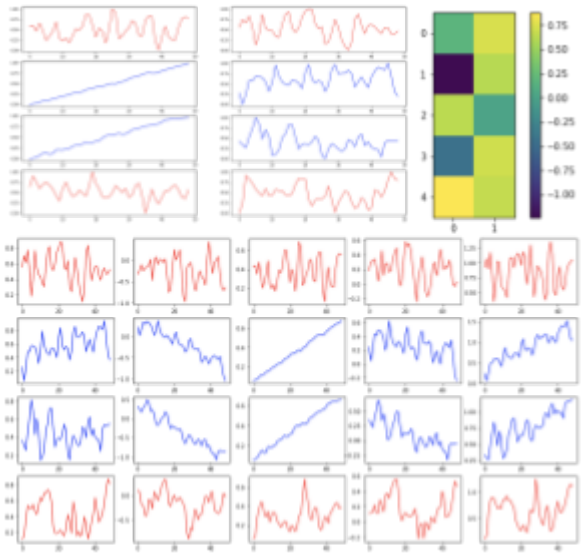

Figure 9: Top: Few sample latent processes (red: negative cases, blue: positive) and mixing matrix, Bottom: Resultant output samples from corresponding processes

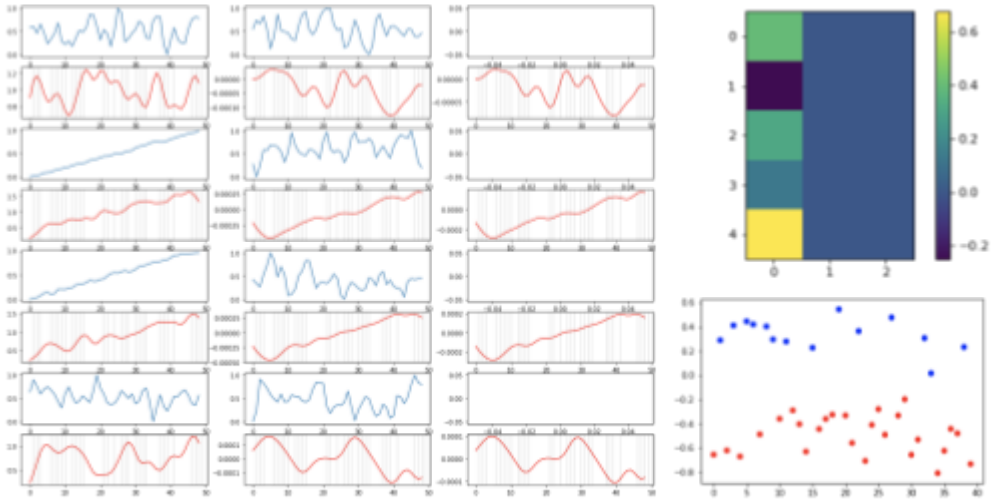

Figure 10: Left: Actual(blue)and extracted(red) latent processes from observed data (vertical lines shows the position of inducing points), Right : Extracted mixing matrix and intermediary variable results on test set

generated mixing matrix to obtain the observation Y and their corresponding labels L (Figure: 9). The observations are then split into training data (Y, L) and test data (Y^*, L^*).

To keep the inference simple, we used only Gaussian kernels in our GP priors with signal noise ($\sigma = 1$). Figure contains the few extracted latent processes for visual inspection, though we use only 60 percent of data, we can see that model is able to extract linear trend from the samples. Rest of the latent processes are generally ignored due to our assumption of noise to be 1 ($\sigma^2 = 1$), which is also the scale of our observed values. Figure 10 shows the separation of training data in latent space. As can be observed, model is able to achieve a good enough separation of test data in latent space.

Table 1: Results of LGPC under different number of output dimensions

	F1-Score		Accuracy		Reconstruction errors	
Output Dimension	Mean	Std	Mean	Std	Training	Test
2	0.95	0.065	0.96	0.046	21.8	22.03
3	0.97	0.032	0.98	0.024	32.71	33.03
5	0.97	0.031	0.98	0.027	51.39	51.62
7	0.98	0.019	0.98	0.012	65.54	65.27
20	0.98	0.015	0.98	0.012	80.02	141.31
50	0.96	0.032	0.97	0.024	249.86	253.74
100	0.98	0.019	0.98	0.019	417.54	402.01
150	0.95	0.05	0.96	0.046	540.95	529.99

5 Experiments and Results

We tested LGPC model in two different experimental settings. This section presents the details of experiments and their results. First, we test the impact of different parameters on classification performance using artificial dataset generated through similar procedure as in previous sections. Later, we use synthetic control time series dataset available at UCI repository [M. Lichman 2013] as a benchmark dataset to compare the classification performance of our model against some fundamental classification models. In a classification setting specially in the scenarios with huge mismatch in number of instances that belong to different categories, accuracy does not provide complete picture and hence we also report F1-Scores as the measure of model's performance. Reconstruction errors are another important metric which when compared between training and test instances can give us an approximate overview of over-fitting in the model and thus are duly reported for the first experiment set.

5.1 Artifical data set

In most of the classification scenarios with multi-series data, number of dimensions in observations (number of sensors for example in sensor data) and sampling resolution (number of samples per sensor) are some important factors that might impact predictive performance of models. In this experiment we use an artificial generated dataset to test LGPC's performance with varying values of output dimension (C) and sampling resolution (N) in datset. Furthermore, we also test model's performance with different number of inducing variable to test model's scalability. We generated the dataset by mixing two latent Gaussian processes. For every sample $s \in S$, we mixed either two gaussian processes or one linear and one gaussian process with 40 percent probability. S scaled samples are then generated by keeping mixing matrix constant throughout the samples. The labels are assigned according to the mixing, i.e. +1 when one of the process has a positive linear slope and -1 otherwise.

5.1.1 Effect of output dimensions

To test the impact of output dimensions, we keep number of latent processes constant (P=2) but change mixing matrix corresponding to a different number of output dimensions.

Table 2: Results of LGPC with varying number of samples in output

	F1-Score		Accuracy		Reconstruction errors	
Number of samples	Mean	Std	Mean	Std	Training	Test
10	0.72	0.087	0.8	0.046	8.9	9.04
20	0.9	0.073	0.92	0.045	16.06	16.27
30	0.87	0.132	0.9	0.087	25.11	25.06
50	0.86	0.224	0.93	0.107	31.24	31.09
100	0.85	0.282	0.92	0.148	63.61	63.78
200	0.98	0.021	0.98	0.012	117.74	119.21
300	0.92	0.096	0.94	0.085	165.094	167.17

Table 3: Results of LGPC under different levels of induction ratio

	F1-Score		Accuracy		Reconstruction errors	
Induction Ratio	Mean	Std	Mean	Std	Training	Test
0.2	0.88	0.24	0.94	0.12	115.75	115.92
0.6	0.96	0.08	0.98	0.02	62.01	61.91
0.8	0.98	0.03	0.99	0.02	64.54	65.56

Fig 1 shows our model's performance with the artificial dataset for different values of C. It's easy to see that model's performance remain pretty stable even when number of output dimension is near or more than the number of samples. Since model projects data back to a much tighter latent and removes much of redundant columns due to prior on mixing matrix favoring smaller values, number of output dimensions end up losing it's relevance, giving pretty much stable performance over increased number of output dimensions.

5.1.2 Number of samples

Fig 2 shows us the results when sampling resolution is varied for fixed number of output dimensions (C=3 in this example). Once again we see that model is robust with in a large range of sampling resolution. Even 20 percent of maximum possible samples are enough to yield good enough f1-scores. Predictive power of course increases as the number of samples available to the model becomes higher.

5.1.3 Inducing points

Finally, we also tested classification power of model with different number of inducing points, The ability to use lesser number of inducing points with minimum impact on performance makes the model scalable since sparse solution drastically reduces the time complexity of inference procedure from $O(N^3)$ to $O(Nn^3)$, where $n \ll N$. As evident from the Fig 3, even 50 percent reduction in the number of points is able to achieve good enough classification accuracy on the dataset indicating that huge speed ups are possible with presented sparse solution of the model.

5.2 Synthetic control dataset

We use a benchmark dataset namely 'synthetic control data set' [1] from UCI to test the classification performance of our model and then comparing it against other

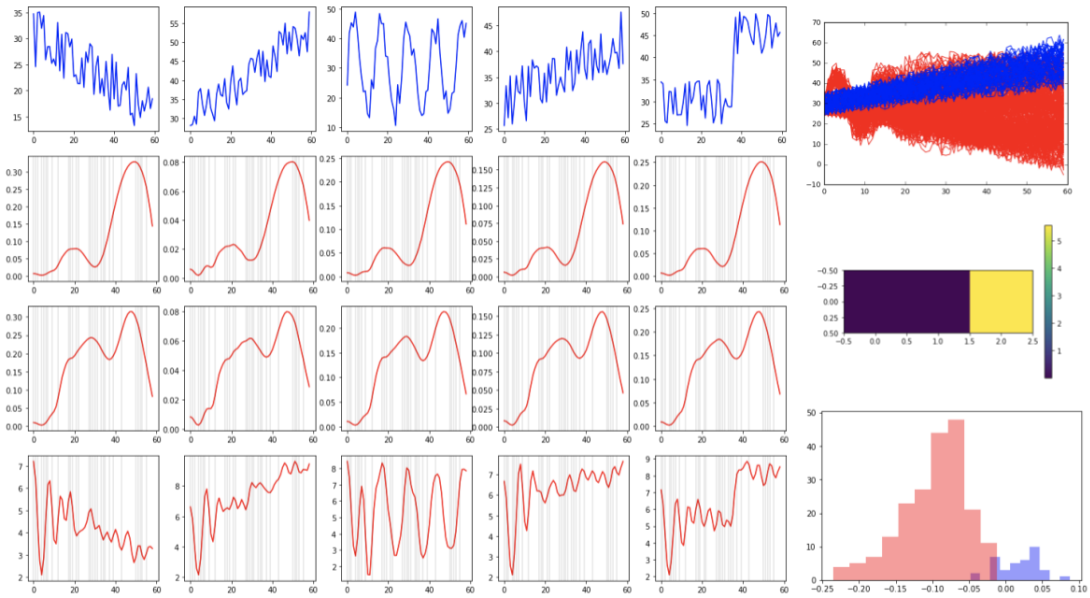

Figure 11: LGPC results on synthetic control dataset: Left: Top rows shows the samples from test dataset, Bottom three rows show respective guessed latent processes. Right top: training dataset with blue showing postive instances, middle: Inferred mixing matrix, bottom: Separation inferred among test instances.

Table 4: Comparative results of different classification methods on synthetic control dataset

Methods	F1-Score	Accuracy
LCGC	0.89	0.97
SVM	0.88	0.97
Logistic Regression	0.88	0.96
LDA	0.9	0.97

standard classification models. The dataset has a collection of 600 series instances with five different trends: Upward and downward trend, upward and downward shift, cyclic trend and random noise. For the experiment we divide entire datset into training and test instances and try to predict the correct classification for test data using the parameters learned during training. For this experiment, LGPC model was employed with it's default settings (P=3, gaussian Kernels) and 0.8 induction ratio. Top right side of the Figure displays the training dataset with positive classes colored in blue. Rest of the figure shows the result of classification; mixing matrix in top middle and the guessed latent processes (2-4th row)for few samples of test instances (first row) in the bottom. Top right shows the clear separation between most of the latent values $l*$ of test instances. As we can observe guesses latent processes of LGPC accurately mimic the actual instance thereby decently separating the test instances into their respective classes. Moreover, Figure 4 shows the comparison between LDA, logistic regression and linear SVM with LGPC, and we can see that even with 0.8 induction ratio and only default settings performance of LGPC is at

par with other time tested algorithms. All of the other classification models were taken from publicly available Scipy implementations[Eric Jones et al. 2001].

5.3 Conclusion

Results on artificially generated dataset were promising and indicate that the model's result will be stable under varying conditions. Classification experiment results using synthetic control dataset were also on par with other state of the art classification algorithms, specially considering that a very simple implementation of LGPC was used. Hyperparameter optimization and inducing point selection schemes can be used to further improve model performance. Only caveat during experimentation is that due to the variational nature of model a bad initialization of parameters might stop model from converging within specified number of iterations. These events are however rare and overall results of all of our experiments indicate that LGPC can be a promising model in cases where the data is generated through a long series of multiple measurements like medical, meteorological settings etc.

6 Summary

In this section we briefly review the body of literary work directly adjacent to our contribution and later summarize the results and conclusions of this thesis. Finally we also mention few immediate pointers to the future works this thesis might eventually lead.

6.1 Related Work

Model proposed in this thesis assumes that the observations as well as their labels were generated by a linear combination of multiple latent processes. [Luttinen 2009b] used a similar approach in probabilistic factor analysis to model spatio-temporal datasets by using Gaussian process priors for both mixing matrix as well as the latent components corresponding to time signals. [Wilson et al. 2012] in their model Gaussian Process Regression Network (GPRN) combine the ideas of Bayesian neural network structure by making the mixing matrix input dependent Gaussian process and thereby introducing non linearity in the mixture as well and created a general purpose multi tasking framework. [Remes et al. 2017] further extends the idea by proposing a novel non stationary Latent Correlational Kernel that creates a shared correlation structure between the input dependent latent processes. Both regression and classification tasks on multi observational sets are handled in [Remes et al. 2017].

Another direction of research is the introduction of more hierarchy in the latent part of the model. [Damianou et al. 2013] proposes an extension to GP-latent variable model which can scale GPLVM vertically by encoding multiple layers of latent spaces as well as horizontally by introducing additional conditional in-dependencies between latent variables.

6.2 Conclusion

To summarize, our main objective in this thesis was to present a latent Gaussian processes classifier that is capable of ingesting multivariate data streams for each instance and categorize them in separate classes by extracting the relevant information to a low dimensional latent space. Gaussian processes enable a rich non linear latent space for the data to be projected into, while a sparse solution alleviates the scalability issues that may creep into when using GPs.

Proposed model was tested as a single class classifier in two different experiment settings, first using an artificial dataset where positive instances consisted of a linear trend mixed with other Gaussian trend to test model's performance with different parameter scenarios like varying amount of output dimensions, induction ratio etc. The results presented in section 3.3.1 of this experiment conclude that the model is stable and classifier performance was consistent. Second experiment was performed using an open source benchmark dataset for time series from UCI, to test model's performance in a more real setting and compare it with other classification methods. Results of this experiment as explained in section 3.3.2 were also very promising and demonstrated that the performance of LGPC with a default settings was comparable

to the open source implementation of other state of the art classification methods like logistic regression, SVM etc. Moreover, being a fully Bayesian model, using LGPC also means that uncertainty information is readily available at every stage of inference be it latent processes or latent predicted values something that's not possible with above mentioned models.

In addition to proposing LGPC, we also extended SLFM model Seeger, Teh, et al. 2004 by enabling it to handle multi-observational scenarios such that each observation has it's own latent space and proposing a sparse variational approximation both to original model as well as our extension of it. As explained in section 1.3, computational complexity of inference in Gaussian process grows with an order of three which can make working with large dataset challenging. Sparse solution thus makes the model more scalable and efficient. Visual demonstrations of these models on artificially generated datasets provided in section 2.2.3 and 2.3.2 respectively indicate that the sparse models are able to satisfactorily capture the latent processes of data even with a 0.5 to 0.7 induction ratio.

The experiments and demonstrations in this thesis were performed only on artificially generated or simulated datasets, hence an immediate future continuation of this work might include the tests and experiments based on real world datasets. Apart from classical fMRI, share recognition datasets many new datasets should be expected following the recent explosion of data collection in IoT as well as wearable device and medical areas. Moreover, all of the experiments the naive LGPC model was used with default Gaussian kernels for latent processes. In real world however, there is always much more information about the supposed latent structure is available. This information can be encoded in the model by using different types of Kernel in the latent processes, for example a periodic or Brownian model might provide better results if the data is known to have cycles or trends structures. Similarly, performing hyper-parameter optimization before variational algorithm should also help improve the performance of LGPC model in real world data analysis problems.

References

Anderson, T W (1956). "Statistical Inference in Factor Analysis". In: V.

Andrieu, Christophe et al. (2003). "An Introduction to MCMC for Machine Learning". In: *Machine Learning* 50.1/2, pp. 5–43.

Attias, Hagai (2000). "A variational Bayesian framework for graphical models". In: *Advances in neural information processing systems*, pp. 209–215.

Bartholomew, David J. et al. (2011). *Latent variable models and factor analysis : a unified approach.* Wiley, p. 277.

Beal, Matthew J. (2003). "Variational Algorithms for Approximate Bayesian Inference". PhD thesis. Gatsby Computational Neuroscience Unit, University College London.

Bishop, Christopher M (1999). "Variational Principal Components". In: *ICANN* 1, pp. 509–514.

Bishop, Christopher M. (2006). *Pattern recognition and machine learning.* Springer, p. 738.

Blei, David M. et al. (2017). "Variational Inference: A Review for Statisticians". In: *Journal of the American Statistical Association* 112.518, pp. 859–877.

Blight, B. J. N. et al. (1975). "A Bayesian Approach to Model Inadequacy for Polynomial Regression". In: *Biometrika* 62.1, p. 79.

Carlin, Bradley P et al. (1995). "Bayesian model choice via Markov chain Monte Carlo methods". In: *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 473–484.

Csató, Lehel et al. (2002). "Sparse On-Line Gaussian Processes". In: *Neural Computation* 14.3, pp. 641–668.

Damianou, Andreas et al. (2013). "Deep Gaussian Processes". In: *Artificial Intelligence and Statistics.* Pp. 207–215.

Dempster, A. P. et al. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39, pp. 1–38.

Eric Jones et al. (2001). *{SciPy}: Open source scientific tools for {Python}.*

Ferguson, Thomas S. (1973). "A Bayesian Analysis of Some Nonparametric Problems". In: *The Annals of Statistics* 1.2, pp. 209–230.

Gelman, Andrew, John B Carlin, et al. (2014). *Bayesian data analysis.* 2nd ed. CRC press Boca Raton, FL.

Gelman, Andrew and Xiao-Li Meng. *Simulating Normalizing Constants: From Importance Sampling to Bridge Sampling to Path Sampling.* Vol. 13. Institute of Mathematical Statistics, pp. 163–185.

Ghahramani, Zoubin (2013). "Bayesian nonparametrics and the probabilistic approach to modelling". In: *The Royal Society* 371.Phil. Trans. R. Soc. A, p. 20110553.

Ghahramani, Zoubin and Matthew J. Beal (2000). "Variational Inference for Bayesian Mixtures of Factor Analysers". In: *Advances in neural information processing systems.* 12, pp. 449–455.

Girolami, Mark (2001). "A Variational Method for Learning Sparse and Overcomplete Representations". In: *Neural Computation* 13.11, pp. 2517–2532.

Grimmett, Geoffrey. et al. (2001). *Probability and random processes*. Oxford University Press, p. 596.

Herbrich, Ralf et al. (2003). "Fast Sparse Gaussian Process Methods: The Informative vector machine". In: *Advances in neural information processing systems*.

Hjort, Nils Lid. (2010). *Bayesian nonparametrics*. Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, p. 299.

Hyvärinen, Aapo et al. (2004). *Independent Component Analysis*. John Willy & Sons.

Jaakko Riihimäki. "Approximate Bayesian Inference". PhD thesis.

Jaakkola, T et al. (1997). "A variational approach to Bayesian logistic regression models and their extensions". In: *Sixth International Workshop on Artificial Intelligence and Statistics* 82, p. 4.

Jolliffe, I. T. (2002). "Principal Component Analysis and factor analysis". In: *Principal Component Analysis*. Springer Series in Statistics. New York: Springer New York.

Jordan, Michael I et al. (1999). "An Introduction to Variational Methods for Graphical Models". In: *Machine Learning* 37, pp. 183–233.

Jöreskog, K. G. (1967). "Some contributions to maximum likelihood factor analysis". In: *Psychometrika* 32.4, pp. 443–482.

Kearns, Michael S. et al. (1999). *Advances in neural information processing systems 11 : proceedings of the 1998 conference*. MIT Press, p. 1090.

Kendall, Maurice G. (Maurice George) et al. (1994). *Kendall's advanced theory of statistics*. Edward Arnold.

Kohavi, Ron (1995). "A study of cross-validation and bootstrap for accuracy estimation and model selection". In: *Ijcai* 14, pp. 1137–1145.

Kohonen, Teuvo (1998). "The self-organizing map". In: *Neurocomputing* 21.1-3, pp. 1–6.

Kolmogoroff, A (1941). "Interpolation und Extrapolation von stationären zufälligen Folgen .(Russian. German - A Web Based Annotated Bibliography - Aigaion 2.0". In: *Bull. Acad. Sci. URSS, Ser. Math.* 5, pp. 3–14.

Krige, D G (1951). "A statistical approach to some mine valuation and allied problems on the Witwatersrand". PhD thesis. University of the Witwatersrand.

Kuss, Malte et al. (2005). "Assessing approximations for Gaussian process classification". In: *Advances in Neural Information Processing Systems*, pp. 699–706.

Lawley, D. N. (1940). "The Estimation of Factor Loadings by the Method of Maximum Likelihood." In: *Proceedings of the Royal Society of Edinburgh* 60.01, pp. 64–82.

Lawrence, Neil D (2004). "Gaussian process latent variable models for visualisation of high dimensional data". In: *Advances in neural information processing systems*. Pp. 329–336.

Lawrence, Neil D. (2005). "Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models". In: *Journal of Machine Learning Research* 6, pp. 1783–1816.

Leonard, Tom (1978). "Density Estimation, Stochastic Processes and Prior Information". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 40, pp. 113–146.

Luttinen, Jaakko (2009a). "Gaussian-process factor analysis for modeling spatio-temporal data". PhD thesis. Aalto University.

— (2009b). "Variational Gaussian-process factor analysis for modeling spatio-temporal data". In: *Advances in neural information processing systems.* Pp. 305–320.

Luttinen, Jaakko et al. (2014). "Linear State-Space Model with Time-Varying Dynamics". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Pp. 338–353.

M. Lichman (2013). *UCI Machine Learning Repository.*

MacKay, David J. C. (1992). "Bayesian Interpolation". In: *Neural Computation* 4.3, pp. 415–447.

— (2003). *Information theory, inference, and learning algorithms.* Cambridge University Press, p. 628.

MacKay, David JC (1994). "Bayesian nonlinear modeling for the prediction competition." In: *ASHRAE transactions* 100.2, pp. 1053–1062.

Matheron, G. (1973). "The Intrinsic Random Functions and Their Applications". In: *Advances in Applied Probability* 5.3, p. 439.

Minka, Thomas P. (2001). "Expectation Propagation for approximate Bayesian Inference". In: *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pp. 362–369.

Minka, Thomas Peter (2001). "A family of algorithms for approximate Bayesian inference". PhD thesis. Massachusetts Institute of Technology,

Murphy, Kevin P. (2012). *Machine learning : a probabilistic perspective.* MIT Press, p. 1067.

Neal, Radford M. (2000). "Markov Chain Sampling Methods for Dirichlet Process Mixture Models". In: *Journal of Computational and Graphical Statistics* 9.2, pp. 249–265.

Neal, Radford M (2001). "Annealed importance sampling". In: *Statistics and computing* 11.2, pp. 125–139.

Nielsen, Frederik Brink (2004). "Variational Approach to Factor Analysis and Related Models". PhD thesis.

O'Hagan, A. et al. (1978). "Curve Fitting and Optimal Design for Prediction". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 40, pp. 1–42.

Orbanz, Peter (2009). "Construction of Nonparametric Bayesian Models from Parametric Bayes Equations". In: *Advances in Neural Information Processing Systems* 22, pp. 1392–1400.

Papoulis, Athanasios et al. (2002). *Probability, random variables, and stochastic processes.* McGraw-Hill, p. 852.

Parisi, Giorgio. (1998). *Statistical field theory.* Perseus Books, p. 352.

Peterson, C. et al. (1987). "A Mean Field Theory Learning Algorithm for Neural Networks by Carsten Peterson and James R. Anderson". In: *Complex Systems* 1, pp. 995–1019.

Press, S. James et al. (1989). "Bayesian Inference in Factor Analysis". In: *Contributions to Probability and Statistics*. New York, NY: Springer New York, pp. 271–287.

Quiñonero-Candela, Joaquin et al. (2001). "A Unifying View of Sparse Approximate Gaussian Process Regression". In: *The Journal of Machine Learning Research* 6, pp. 1939–1959.

Quintana, Fernando A. et al. (2004). "Nonparametric Bayesian Data Analysis". In: *Statistical Science* 19.1, pp. 95–110.

Raiko, Tapani et al. (2007). "Building blocks for variational Bayesian learning of latent variable models". In: *Journal of Machine Learning Research*.

Rasmussen, Carl Edward (1999). "The Infinite Gaussian Mixture Model". In: *Advances in Neural Information Processing Systems 12*. Ed. by S A Solla et al., pp. 554–560.

Rasmussen, Carl Edward and Christopher KI Williams. (2006). *Gaussian process for machine learning*. Vol. 1.

Rasmussen, Carl Edward and Zoubin Ghahramani (2001). "Occam's razor". In: *Advances in neural information processing systems*, pp. 294–300.

Remes, Sami et al. (2017). "Latent Correlation Gaussian Processes". In: *To appear in Asian Conference on Machine Learning*.

Ross, Sheldon M. (1996). *Stochastic processes*. 2nd ed. Wiley series in probability and statistics. Wiley, p. 510.

Rowe, Daniel B. et al. (1998). "Gibbs Sampling and Hill Climbing in Bayesian Factor Analysis". In:

Roweis, Sam et al. (1999). "A Unifying Review of Linear Gaussian Models". In: *Neural Computation* 11.2, pp. 305–345.

Rubin, Donald B. et al. (1982). "EM algorithms for ML factor analysis". In: *Psychometrika* 47.1, pp. 69–76.

Scholz, Matthias et al. (2008). "Nonlinear Principal Component Analysis: Neural Network Models and Applications". In: *Principal manifolds for data visualization and dimension reduction*. Springer, pp. 44–67.

Seeger, Matthias (2004). "Gaussian process for machine learning". In: *International Journal of Neural Systems* 14.02, pp. 69–106.

Seeger, Matthias, Yee-whye Teh, et al. (2004). "Semiparametric latent factor models". In: *No. EPFL-REPORT-161465. 2005.*

Seeger, Matthias, Christopher K. I. Williams, et al. (2003). "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression". In: *Artificial Intelligence and Statistics*. 9.

Shawe-Taylor, John. et al. (2004). *Kernel methods for pattern analysis*. Cambridge University Press, p. 462.

Snelson, Edward Lloyd (2008). "Flexible and efficient Gaussian process models for machine learning". PhD thesis. University College of London.

Snelson, Edward et al. (2006). "Sparse Gaussian processes using pseudo-inputs". In: *Advances in neural information processing systems*. 18, pp. 1257–1264.

Spearman, C. (1904). "General Intelligence; Objectively Determined and Measured". In: *The American Journal of Psychology* 15.2, p. 201.

Tipping, Michael E. et al. (1999). "Probabilistic Principal Component Analysis". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3, pp. 611–622.

Titsias, Michalis (2009). "Variational Learning of Inducing Variables in Sparse Gaussian Processes". In: *Conference on Artificial Intelligence and Statistics* 5.Proceedings of Machine Learning Research, pp. 567–574.

Titsias, Michalis K (2009). "Variational Model Selection for Sparse Gaussian Process Regression". In: *Technical report*.

Tokdar, Surya T. "Towards a Faster Implementation of Density Estimation with Logistic Gaussian Process Priors". In: *Journal of Computational and Graphical Statistics* 16, pp. 633–655.

Tolvanen, Ville et al. (2014). "Expectation propagation for nonstationary heteroscedastic Gaussian process regression". In: *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, pp. 1–6.

Wiener, Norbert et al. (1964). *Extrapolation, interpolation, and smoothing of stationary time series with engineering applications*. Technology Press of the Massachusetts Institute of Technology.

Williams, Christopher KI et al. (1996). "Gaussian processes for regression". In: *Advances in neural information processing systems*, pp. 514–520.

Williams, C.K.I. et al. (1998). "Bayesian classification with Gaussian processes". In: *EEE Transactions on Pattern Analysis and Machine Intelligence* 20.12, pp. 1342–1351.

Williams et al. (1998). "Prediction with Gaussian processes: from linear regression to linear prediction and beyond". In: *Proceedings of the NATO Advanced Study Institute on Learning in graphical models*. Kluwer Academic Publishers, pp. 599–621.

Wilson, Andrew Gordon et al. (2012). "Gaussian Process Regression Networks". In: *International Conference for Machine Learning*.

Yu, Byron M. et al. (2008). "Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity". In: *Advances in Neural Information Processing Systems* 21, pp. 1881–1888.

Zhao, Jian-Hua et al. (2009). "A note on variational Bayesian factor analysis". In: *Neural Networks* 22, pp. 988–997.

A Details of GP based factor analysis posterior inference

In order to make the posterior computationally tractable, we introduce another distribution $\hat{u_p}$ which is evaluated over fewer values than the original u_p. Thus posterior distribtion $p(\phi, u, \hat{u} \mid Y)$ is approximated as a factorized distribution given by:

$$q(\phi, u, \hat{u}) = q(\phi) \prod_{p=1}^{P} p(u|\hat{u_p}) q(\hat{u_p})$$

After introducing sparse variational approximations and taking log both sides of marginal likelihood of the observations, equation 24 can be written as,

$$L(Y|X) = \int q(\phi) \prod_{p=1}^{P} p(u_p|\hat{u_p}) q(\hat{u}_p) \log \frac{p(Y|\hat{u}, \phi) p(\phi) \prod_{p=1}^{P} p(\hat{u_p})}{q(\phi) \prod_{p=1}^{P} q(\hat{u_p})} d\phi d\hat{u} \tag{A1}$$

rewriting above equation in terms of $\hat{u_p}$,

$$L(\hat{u}_p) = \int q(\hat{u}_p) log \frac{\bar{p}(Y|\phi, \hat{u}_p) p(\hat{u}_p)}{\hat{u}_p} \tag{A2}$$

where,

Now, we can factorizing Y over it's row, i.e. C's,

$$\log \bar{p}(Y|\phi, \hat{u}_p) = \sum_{c}^{C} < log(Y_c|\phi_c, u) >_{\phi, p_{1..P \setminus p}, u_p|\hat{u}_p} - KLs \tag{A3}$$

After expanding the terms and completing the square we can obtain following parameterization:

$$\log \bar{p}(Y|\phi, \hat{u}_p) = \log N(S_p^{-1} z_p | K_{Nn} K_n^{-1} \hat{u}_p, K_N - K_{Nn} K_n^{-1} K_{nN}) - \frac{1}{2} tr(\sum_{i}^{P} S_i * cov(u_i|\hat{u_i})) - KLs \tag{A4}$$

where

$$z_p = \sum_{c}^{C} \mathbb{E}[\phi_{cp}](y_c - \sum_{i}^{P/p} \mathbb{E}[\phi_{ci}] \mathbb{E}[u_{ip}]),$$

and

$$S_p = \sum_{c}^{C} \mathbb{E}[\phi_{cp}^2]$$

Substituting this back in the equation A2,

$\mathrm{L}_{(p)} = \int q(\hat{u_p}) \log \frac{N(S_p^{-1} z_p | K_{Nn} K_n^{-1} \hat{u}_p, K_{NN} - K_{Nn} K_{nn}^{-1} K_{nN}) p(\hat{u}_p)}{q(u_p)} - \frac{1}{2} tr(\sum_i^P S_i * cov(u_i \mid \hat{u_i})) - KLs$

Above equation can be used for hyperparmater selection however to find the optimal distribution $q(\hat{u_p})$ one can discard the constant terms and use only the lower

bound:

$$L(\hat{u}_p) \geq \int q(\hat{u_p}) \log \frac{N(S_p^{-1} z_p \mid K_{Nn} K_n^{-1} \hat{u}_p, K_{NN} - K_{Nn} K_{nn}^{-1} K_{nN}) p(\hat{u}_p)}{q(u_p)} \tag{A5}$$

The ELBO obtained above can be interpreted as the KL distance between variational distribution of $\hat{u_p}$ and the nominator value. This distance naturally is minimal when denominator $\hat{u_p}$ takes the value same as the nominator, i.e.

$$q(\hat{u}_p) \approx N(S_p^{-1} z_p \mid K_{Nn} K_n^{-1} \hat{u}_p, K_{NN} - K_{Nn} K_{nn}^{-1} K_{nN}) p(\hat{u}_p) \tag{A6}$$

Rearranging terms

$$q(\hat{u}_p) \approx N(\Sigma_p^{-1} K_{nn}^{-1} K_{Nn} z_p, \Sigma_p^{-1})$$

where

$$\Sigma_p = K_n^{-1} + \frac{1}{\sigma^2} K_n^{-1} K_{nN} S_p K_{Nn} K_n^{-1} \tag{A7}$$

Note: More rigorous ways to reach at identical optimal distribution using variational calculus or by reversing Jensen's equality can be found in [https://arxiv.org/pdf/1402.1412.pdf and [Tistia 09 tech report] respectively.

Moreover, we know that ,

$$q(u_p) = \int p(u_p \mid \hat{u_p}) q(\hat{u_p}) d\hat{u_p}$$

where factor $p(u_p \mid \hat{u_p})$ can be easily obtained through conditional of GP priors u_p and $\hat{u_p}$. Using affine property of gaussians [Bishop pg 87-89] and equation A we can obtain,

$$q(u_p) \approx \mathcal{N}(u_p \mid M \hat{\mu_p}, \Sigma_{u|\hat{u}}^p + M \Sigma_p M^T)$$

where $\hat{\mu_p}$ is the mean of GP $\hat{u_p}$, $M = K_{Nn} K_{nn}^{-1}$ and $\Sigma_{u|\hat{u}}^p = K_{NN} - M K_{nN}$

An optimization process similar to $\hat{u_p}$ can be followed to find the lower bound with respect to ϕ,i.e.

$$L(\phi) = \int q(\phi) \log \frac{\bar{p}(Y|\phi, u) p(\phi)}{q(\phi)} d\phi \tag{A8}$$

where

$$\log \bar{p}(Y|\phi, u) = \int q(u) log \frac{p(Y|\phi, u) p(u)}{q(u)} du = \int q(u) \log p(Y|\phi, u) du - KL(q(u)||p(u))$$

Variational distribution is given by,

$$q(\phi) \approx \mathcal{N}(\phi \mid y \mathbb{E}[U]^T \Sigma_\phi^{-1}, \Sigma_\phi^{-1})$$

where $\Sigma_\phi = (V_\phi^{-1} + I)$ and $V_\phi = \mathbb{E}[u] \mathbb{E}[u]^T \sigma^2$